# Package 'phylogram'

June 15, 2017

**Type** Package

**Title** Dendrograms for Evolutionary Analysis

**Version** 1.0.1

**Author** Shaun Wilkinson [aut, cre]

**Maintainer** Shaun Wilkinson <shaunpwilkinson@gmail.com>

**Description** Contains functions for importing and exporting 'dendrogram' objects in parenthetic text format, and several functions for command-line tree manipulation. With an emphasis on speed and computational efficiency, the package also includes a suite of tools for rapidly computing distance matrices and building large trees using fast alignment-free 'k-mer' counting and divisive clustering techniques.

**License** GPL-3

**LazyData** TRUE

**URL** http://github.com/shaunpwilkinson/phylogram

**BugReports** http://github.com/shaunpwilkinson/phylogram/issues

**Encoding** UTF-8

**Depends** R (>= 3.0.0)

**Imports** openssl, Rcpp (>= 0.12.8), stats,

**Suggests** ape (>= 3.0), knitr, rmarkdown, testthat

**LinkingTo** Rcpp

**RoxygenNote** 6.0.1

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2017-06-14 23:58:19 UTC

# R topics documented:



---

`kcount` *K-mer counting.*

---

## Description

Count all k-letter words in a sequence or set of sequences with a sliding window of length k.

## Usage

```
kcount(x, k = 5, residues = NULL, gap = "-")
```

## Arguments

| | |
|---|---|
| `x` | a matrix of aligned sequences, a list of unaligned sequences, or a vector representing a single sequence. Accepted modes are "character" and "raw" (the latter being applicable for "DNAbin" and "AAbin" objects). |
| `k` | integer representing the k-mer size. Defaults to 5. Note that high values of k may be slow to compute and use a lot of memory due to the large numbers of calculations required, particularly when the residue alphabet is also large. |
| `residues` | either NULL (default; the residue alphabet is automatically detected from the sequences), a case sensitive character vector specifying the residue alphabet, or one of the character strings "RNA", "DNA", "AA", "AMINO". Note that the default option can be slow for large lists of character vectors. Specifying the residue alphabet is therefore recommended unless x is a "DNAbin" or "AAbin" object. |
| `gap` | the character used to represent gaps in the alignment matrix (if applicable). Ignored for `"DNAbin"` and `"AAbin"` objects. Defaults to "-" otherwise. |

## Details

This function computes a vector or matrix of k-mer counts from a sequence or set of sequences using a sliding a window of length k. DNA and amino acid sequences can be passed to the function either as a list of non-aligned sequences or a matrix of aligned sequences, preferably in the "DNAbin" or "AAbin" raw-byte format (Paradis et al 2004, 2012; see the ape package documentation for more information on these S3 classes). Character sequences are supported; however ambiguity codes may not be recognized or treated appropriately, since raw ambiguity codes are counted according to their underlying residue frequencies (e.g. the 5-mer "ACRGT" would contribute 0.5 to the tally for "ACAGT" and 0.5 to that of "ACGGT").

To minimize computation time when counting longer k-mers (k > 3), amino acid sequences in the raw "AAbin" format are automatically compressed using the Dayhoff-6 alphabet as detailed in Edgar (2004). Note that amino acid sequences will not be compressed if they are supplied as a list of character vectors rather than an "AAbin" object, in which case the k-mer length should be reduced (k < 4) to avoid excessive memory use and computation time.

## Value

Returns a matrix of k-mer counts with one row for each sequence and *n^k* columns (where *n* is the size of the residue alphabet and *k* is the k-mer size)

## Author(s)

Shaun Wilkinson

## References

Edgar RC (2004) Local homology recognition and distance measures in linear time using compressed amino acid alphabets. *Nucleic Acids Research*, **32**, 380-385.

Paradis E, Claude J, Strimmer K, (2004) APE: analyses of phylogenetics and evolution in R language. *Bioinformatics* **20**, 289-290.

Paradis E (2012) Analysis of Phylogenetics and Evolution with R (Second Edition). Springer, New York.

## See Also

kdistance for k-mer distance matrix computation.

## Examples

```
## compute a matrix of k-mer counts for the woodmouse
## data (ape package) using a k-mer size of 3
library(ape)
data(woodmouse)
x <- kcount(woodmouse, k = 3)
x
## 64 columns for nucleotide 3-mers AAA, AAC, ... TTT
## convert to AAbin object and repeat the operation
y <- kcount(ape::trans(woodmouse, 2), k = 2)
y
```

```
## 400 columns for amino acid 2-mers AA, AB, ... , YY
```

---

kdistance *K-mer distance matrix computation.*

---

## Description

Computes the matrix of k-mer distances between all pairwise comparisons of a set of sequences.

## Usage

```
kdistance(x, k = 5, method = "edgar", residues = NULL, gap = "-", ...)
```

## Arguments

| | |
|---|---|
| x | a matrix of aligned sequences or a list of unaligned sequences. Accepted modes are "character" and "raw" (the latter being applicable for "DNAbin" and "AAbin" objects). |
| k | integer representing the k-mer size to be used for calculating the distance matrix. Defaults to 5. Note that high values of k may be slow to compute and use a lot of memory due to the large numbers of calculations required, particularly when the residue alphabet is also large. |
| method | a character string giving the k-mer distance measure to be used. Currently the available options are "edgar" (default; see Edgar (2004) for details) and the standard methods available for the base function "dist" ("euclidean", "maximum", "manhattan", "canberra", "binary" and "minkowski"). |
| residues | either NULL (default; the residue alphabet is automatically detected from the sequences), a case sensitive character vector specifying the residue alphabet, or one of the character strings "RNA", "DNA", "AA", "AMINO". Note that the default option can be slow for large lists of character vectors. Specifying the residue alphabet is therefore recommended unless x is a "DNAbin" or "AAbin" object. |
| gap | the character used to represent gaps in the alignment matrix (if applicable). Ignored for "DNAbin" or "AAbin" objects. Defaults to "-" otherwise. |
| ... | further arguments to be passed to "as.dist". |

## Details

This function computes the $n * n$ k-mer distance matrix (where $n$ is the number of sequences), returning an object of class "dist". DNA and amino acid sequences can be passed to the function either as a list of non-aligned sequences or as a matrix of aligned sequences, preferably in the "DNAbin" or "AAbin" raw-byte format (Paradis et al 2004, 2012; see the ape package documentation for more information on these S3 classes). Character sequences are supported; however ambiguity codes may not be recognized or treated appropriately, since raw ambiguity codes are counted according to their underlying residue frequencies (e.g. the 5-mer "ACRGT" would contribute 0.5 to the tally for "ACAGT" and 0.5 to that of "ACGGT").

To minimize computation time when counting longer k-mers (k > 3), amino acid sequences in the raw "AAbin" format are automatically compressed using the Dayhoff-6 alphabet as detailed in Edgar (2004). Note that amino acid sequences will not be compressed if they are supplied as a list of character vectors rather than an "AAbin" object, in which case the k-mer length should be reduced (k < 4) to avoid excessive memory use and computation time.

## Value

an object of class `"dist"`.

## Author(s)

Shaun Wilkinson

## References

Edgar RC (2004) Local homology recognition and distance measures in linear time using compressed amino acid alphabets. *Nucleic Acids Research*, **32**, 380-385.

Paradis E, Claude J, Strimmer K, (2004) APE: analyses of phylogenetics and evolution in R language. *Bioinformatics* **20**, 289-290.

Paradis E (2012) Analysis of Phylogenetics and Evolution with R (Second Edition). Springer, New York.

## See Also

`kcount` for k-mer counting, and `mbed` for leaner distance matrices

## Examples

```
## compute a k-mer distance matrix for the woodmouse
## dataset (ape package) using a k-mer size of 5
library(ape)
data(woodmouse)
### subset global alignment by removing gappy ends
woodmouse <- woodmouse[, apply(woodmouse, 2, function(v) !any(v == 0xf0))]
### compute the distance matrix
woodmouse.dist <- kdistance(woodmouse, k = 5)
### cluster and plot UPGMA tree
woodmouse.tree <- as.dendrogram(hclust(woodmouse.dist, "average"))
plot(woodmouse.tree)
```

---

`ladder` *Reorder tree branches in ladderized pattern.*

---

## Description

This function ladderizes the branches of a `dendrogram` object to aid in visual interpretation.

## Usage

```
ladder(x, decreasing = FALSE)
```

## Arguments

| | |
|---|---|
| x | an object of class "dendrogram". |
| decreasing | logical indicating whether the tree should be ladderized upwards or downwards. Defaults to FALSE (downwards). |

## Details

This function is the dendrogram analogue of the ladderize function in the ape package (Paradis et al 2004, 2012).

## Value

Returns an object of class dendrogram.

## Author(s)

Shaun Wilkinson

## References

Paradis E, Claude J, Strimmer K, (2004) APE: analyses of phylogenetics and evolution in R language. *Bioinformatics* **20**, 289-290.

Paradis E (2012) Analysis of Phylogenetics and Evolution with R (Second Edition). Springer, New York.

## See Also

The ladderize function in the ape package performs a similar operation for objects of class "phylo".

## Examples

```
x <- read.dendrogram(text = "(A:0.1,B:0.2,(C:0.3,D:0.4):0.5);")
plot(x, horiz = TRUE)
x <- ladder(x, decreasing = TRUE)
plot(x, horiz = TRUE)
```

mbed *Convert sequences to vectors of distances to a subset of seed sequences.*

## Description

This function computes a matrix of distances from each sequence to a subset of 'seed' sequences using the method outlined in Blacksheilds et al (2010).

## Usage

```
mbed(x, seeds = NULL, k = 5, residues = NULL, gap = "-",
  counts = FALSE)
```

## Arguments

| | |
|---|---|
| x | a matrix of aligned sequences or a list of unaligned sequences. Accepted modes are "character" and "raw" (the latter is for "DNAbin" and "AAbin" objects). |
| seeds | optional integer vector indicating which sequences should be used as the seed sequences. If `seeds = NULL` a set of log(*n*, 2)^2 non-identical sequences is randomly selected from the sequence set (where *n* is the number of sequences; see Blacksheilds et al. 2010). Alternatively, if `seeds = 'all'` a standard *n* * *n* distance matrix is computed. |
| k | integer representing the k-mer size to be used for calculating the distance matrix. Defaults to 5. Note that high values of k may be slow to compute and use a lot of memory due to the large numbers of calculations required, particularly when the residue alphabet is also large. |
| residues | either NULL (default; emitted residues are automatically detected from the sequences), a case sensitive character vector specifying the residue alphabet, or one of the character strings "RNA", "DNA", "AA", "AMINO". Note that the default option can be slow for large lists of character vectors. Specifying the residue alphabet is therefore recommended unless x is a "DNAbin" or "AAbin" object. |
| gap | the character used to represent gaps in the alignment matrix (if applicable). Ignored for `"DNAbin"` or `"AAbin"` objects. Defaults to "-" otherwise. |
| counts | logical indicating whether the (usually large) matrix of k-mer counts should be returned as an attribute of the returned object. Defaults to FALSE. |

## Details

This function computes a *n* * log(*n*, 2)^2 k-mer distance matrix (where *n* is the number of sequences), returning an object of class `"mbed"`. If the number of sequences is less than or equal to 19, the full *n* * *n* distance matrix is produced (since the rounded up value of log(*19*, 2)^2 is 19). Currently the only distance measure supported is that of Edgar (2004).

For maximum information retention following the embedding process it is generally desirable to select the seed sequences based on their uniqueness, rather than simply selecting a random subset

(Blackshields et al. 2010). Hence if 'seeds' is set to NULL (the default setting) the the 'mbed' function selects the subset by clustering the sequence set into $t$ groups using the k-means algorithm ($k = t$), and choosing one representative from each group. Users can alternatively pass an integer vector (as in the above example) to specify the seeds manually. See Blackshields et al (2010) for other seed selection options.

DNA and amino acid sequences can be passed to the function either as a list of non-aligned sequences or as a matrix of aligned sequences, preferably in the "DNAbin" or "AAbin" raw-byte format (Paradis et al 2004, 2012; see the ape package documentation for more information on these S3 classes). Character sequences are supported; however ambiguity codes may not be recognized or treated appropriately, since raw ambiguity codes are counted according to their underlying residue frequencies (e.g. the 5-mer "ACRGT" would contribute 0.5 to the tally for "ACAGT" and 0.5 to that of "ACGGT").

To minimize computation time when counting longer k-mers (k > 3), amino acid sequences in the raw "AAbin" format are automatically compressed using the Dayhoff-6 alphabet as detailed in Edgar (2004). Note that amino acid sequences will not be compressed if they are supplied as a list of character vectors rather than an "AAbin" object, in which case the k-mer length should be reduced (k < 4) to avoid excessive memory use and computation time.

Note that agglomerative (bottom-up) tree-building methods such as neighbor-joining and UPGMA depend on a full $n * n$ distance matrix. See the kdistance function for details on computing symmetrical distance matrices.

## Value

Returns an object of class "mbed", whose primary object is an $n * \log(n, 2)^2$ matrix (where $n$ is the number of sequences). The returned object contains additional attributes including an integer vector of seed sequence indices ("seeds"), a logical vector identifying the duplicated sequences ("duplicates"), an integer vector giving the matching indices of the non-duplicated sequences ("pointers"), a character vector of MD5 digests of the sequences ("hashes"), an integer vector of sequence lengths ("seqlengths"), and if counts = TRUE, the matrix of k-mer counts ("kcounts"; see kcount for details).

## Author(s)

Shaun Wilkinson

## References

Blackshields G, Sievers F, Shi W, Wilm A, Higgins DG (2010) Sequence embedding for fast construction of guide trees for multiple sequence alignment. *Algorithms for Molecular Biology*, **5**, 21.

Edgar RC (2004) Local homology recognition and distance measures in linear time using compressed amino acid alphabets. *Nucleic Acids Research*, **32**, 380-385.

Paradis E, Claude J, Strimmer K, (2004) APE: analyses of phylogenetics and evolution in R language. *Bioinformatics* **20**, 289-290.

Paradis E (2012) Analysis of Phylogenetics and Evolution with R (Second Edition). Springer, New York.

## See Also

kdistance for full *n* * *n* distance matrix computation.

## Examples

```
## compute an embedded k-mer distance matrix for the woodmouse
## dataset (ape package) using a k-mer size of 5
library(ape)
data(woodmouse)
## randomly select three sequences as seeds
set.seed(999)
seeds <- sample(1:15, size = 3)
## embed the woodmouse dataset in three dimensions
woodmouse.mbed <- mbed(woodmouse, seeds = seeds, k = 5)
## print the distance matrix (without attributes)
print(woodmouse.mbed[,], digits = 2)
```

---

phylogram *Dendrograms for evolutionary analysis.*

---

## Description

The phylogram package contains functions for importing and exporting dendrogram objects in the Newick parenthetic text format, as well as several functions for command-line tree manipulation. With an emphasis on speed and computational efficiency, the package also includes a suite of tools for rapidly computing distance matrices and building large trees using fast alignment-free k-mer counting and divisive clustering techniques.

## Functions

A breif description of the primary **phylogram** functions are provided with links to their help pages below.

## File import/export

- read.dendrogram is a text parser that converts parenthetic text (Newick strings) into objects of class "dendrogram"
- write.dendrogram outputs an object of class "dendrogram" to a text string or file in Newick/New Hampshire format

## Tree building

- kcount tabulates all of the k-letter words in a sequence or set of sequence
- kdistance calculates pairwise distances between sequences by k-mer counting
- mbed embeds sequences as vectors of distances to a set of 'seed' sequences
- topdown builds a phylogenetic tree by successively splitting a set of sequences (recursive partitioning)

### Tree editing and manipulation

- prune remove branches from a dendrogram object based on regular expression pattern matching
- ladder reorders the branches of a dendrogram object to aid visualization
- remidpoint recursively sets "midpoint" and "members" attributes for a nested list/dendrogram object
- reposition shifts a dendrogram object up or down (or sideways if plotted horizontally)
- ultrametricize modifies the "height" attributes of the nodes such that all leaves terminate at zero

---

`print.mbed` *Print summary methods.*

---

## Description

Print summary methods.

## Usage

```
## S3 method for class 'mbed'
print(x, ...)
```

## Arguments

| | |
|---|---|
| x | object of various classes. |
| ... | additional arguments to be passed between methods. |

## Value

NULL (invisibly)

## Author(s)

Shaun Wilkinson

prune *Remove tree nodes by regular expression pattern matching.*

## Description

"prune" takes an object of class "dendrogram" and removes all branches whose branch labels match a given regular expression.

## Usage

```
prune(tree, pattern, invert = FALSE, untag = FALSE, ...)
```

## Arguments

| | |
|---|---|
| tree | an object of class "dendrogram". |
| pattern | a regular expression. |
| invert | logical indicating whether the branches whose labels match the regular expression provided in "pattern" should be discarded and the others kept (FALSE; default) or vice versa. Nodes without "label" attributes are ignored. |
| untag | logical (used only when invert = TRUE). Indicates whether the specified pattern should be removed from the branch labels in the returned object. |
| ... | further arguments to be passed to grepl and gsub. |

## Details

This function recursively tests the "label" attribute of each dendrogram node (including non-leaf inner nodes if applicable) for the specified pattern, removing those that register a positive hit. Note that positive matching inner nodes are removed along with all of their sub-nodes, regardless of whether the "label" attributes of the sub-nodes match the pattern.

## Value

Returns an object of class "dendrogram".

## Author(s)

Shaun Wilkinson

## See Also

The drop.tip function in the ape package performs a similar operation for objects of class "phylo". See regex for help with compiling regular expressions.

## Examples

```
x <- read.dendrogram(text = "(A:0.1,B:0.2,(C:0.3,D:0.4):0.5);")
plot(x, horiz = TRUE)
x <- prune(x, pattern = "^A$")
plot(x, horiz = TRUE)
```

---

`read.dendrogram` *Read a dendrogram from parenthetic text.*

---

## Description

`read.dendrogram` parses a text file or character string in Newick (New Hampshire) format and creates an object of class `"dendrogram"`.

## Usage

```
read.dendrogram(file = "", text = NULL, edges = TRUE, ...)
```

## Arguments

| | |
|---|---|
| `file` | character string giving a valid path to the file from where to read the data. |
| `text` | optional character string in lieu of a "file" argument. If a text argument is provided instead of a file path, the data are read via a text connection. |
| `edges` | logical indicating whether edge weights provided in the Newick string should be retained in the returned object (defaults to TRUE). |
| `...` | further arguments to be passed to `scan`. |

## Details

There are varying interpretations of the Newick/New Hampshire text format. This function tries to adhere to the Felsenstein standard outlined here. The function supports weighted edges, labels with special metacharacters (enclosed in single quotation marks), comments (enclosed in square brackets; ignored by the parser), multifuricating nodes, and both rooted and unrooted trees. Comments enclosed in square brackets are also discarded. Inner-node labels (for example "(B:6.0,(A:5.0,C:3.0,E:4.0)Ancestor1:5.0,D:11 are also currently ignored; however, the parsing of “label” attributes for non-leaf dendrogram nodes will be available in a future version.

## Value

Returns an object of class `"dendrogram"`.

## Author(s)

Shaun Wilkinson

## References

http://evolution.genetics.washington.edu/phylip/newicktree.html http://evolution.genetics.washington.edu/phylip/newick_doc.html

## See Also

write.dendrogram writes an object of class "dendrogram" to a text string. The read.tree function in the ape package performs a similar operation for objects of class "phylo" and "multiPhylo".

## Examples

```
newick <- "(A:0.1,B:0.2,(C:0.3,D:0.4):0.5);"
x <- read.dendrogram(text = newick)
plot(x, horiz = TRUE)
```

---

# remidpoint *Set dendrogram attributes for a nested list.*

---

## Description

remidpoint is a helper function used for manually creating "dendrogram" objects from nested lists. The function recursively assigns the necessary 'midpoint' and 'members' attributes at each node.

## Usage

```
remidpoint(x)
```

## Arguments

| | |
|---|---|
| x | a nested list, possibly of class "dendrogram" |

## Value

returns a nested list, or an object of class "dendrogram" depending on the class of the input object.

## Author(s)

Shaun Wilkinson

## Examples

```
## manually create a small dendrogram with three members, A, B and C
x <- list("A", list("B", "C"))
attr(x[[1]], "leaf") <- TRUE
attr(x[[2]][[1]], "leaf") <- TRUE
attr(x[[2]][[2]], "leaf") <- TRUE
attr(x[[1]], "label") <- "A"
attr(x[[2]][[1]], "label") <- "B"
attr(x[[2]][[2]], "label") <- "C"
attr(x, "height") <- 1
attr(x[[1]], "height") <- 0
attr(x[[2]], "height") <- 0.5
attr(x[[2]][[1]], "height") <- 0
attr(x[[2]][[2]], "height") <- 0
x <- remidpoint(x)
class(x) <- "dendrogram"
plot(x, horiz = TRUE)
```

---

# reposition *Reset dendrogram height attributes.*

---

## Description

`reposition` is a helper function used for manually creating `"dendrogram"` objects from nested lists. The function recursively reassigns the 'height' attributes at each node by a given constant.

## Usage

```
reposition(x, shift = "reset")
```

## Arguments

| | |
|---|---|
| `x` | an object of class `"dendrogram"`. |
| `shift` | either the character string "reset" (shift the graph so that the height of the farthest leaf from the root is zero), or a numeric value giving the amount to shift the graph along the primary axis. |

## Value

Returns an object of class `"dendrogram"`.

## Author(s)

Shaun Wilkinson

## Examples

```
x <- read.dendrogram(text = "(A:0.1,B:0.2,(C:0.3,D:0.4):0.5);")
plot(x, horiz = TRUE)
x <- reposition(x)
plot(x, horiz = TRUE)
```

---

topdown *Top down (divisive) tree-building.*

---

## Description

Create phylogenetic trees by successively splitting the sequence dataset into smaller and smaller subsets.

## Usage

```
topdown(x, k = 5, residues = NULL, gap = "-", ...)
```

## Arguments

| | |
|---|---|
| x | a list or matrix of sequences, possibly an object of class "DNAbin" or "AAbin". |
| k | integer. The k-mer size required. |
| residues | either NULL (default; emitted residues are automatically detected from the sequences), a case sensitive character vector specifying the residue alphabet, or one of the character strings "RNA", "DNA", "AA", "AMINO". Note that the default option can be slow for large lists of character vectors. Specifying the residue alphabet is therefore recommended unless the sequence list is a "DNAbin" or "AAbin" object. |
| gap | the character used to represent gaps in the alignment matrix (if applicable). Ignored for "DNAbin" or "AAbin" objects. Defaults to "-" otherwise. |
| ... | further arguments to be passed to kmeans (not including centers). |

## Details

This function creates a tree by successively splitting the dataset into smaller and smaller subsets (recursive partitioning). This is a divisive, or "top-down" approach to tree-building, as opposed to agglomerative "bottom-up" methods such as neighbour joining and UPGMA. It is particularly useful for large large datasets with many sequences ($n > 10,000$) since the need to compute a large $n * n$ distance matrix is circumvented. Instead, a matrix of k-mer counts is computed, and split recursively row-wise using a k-means clustering algorithm ($k = 2$). This effectively reduces the time and memory complexity from quadratic to linear, while generally maintaining comparable accuracy.

If a more accurate tree is required, users can increase the value of nstart passed to kmeans *via* the ... argument. While this can increase computation time, it can improve tree accuracy considerably.

DNA and amino acid sequences can be passed to the function either as a list of non-aligned sequences or a matrix of aligned sequences, preferably in the "DNAbin" or "AAbin" raw-byte format

(Paradis et al 2004, 2012; see the ape package documentation for more information on these S3 classes). Character sequences are supported; however ambiguity codes may not be recognized or treated appropriately, since raw ambiguity codes are counted according to their underlying residue frequencies (e.g. the 5-mer "ACRGT" would contribute 0.5 to the tally for "ACAGT" and 0.5 to that of "ACGGT").

To minimize computation time when counting longer k-mers (k > 3), amino acid sequences in the raw "AAbin" format are automatically compressed using the Dayhoff-6 alphabet as detailed in Edgar (2004). Note that amino acid sequences will not be compressed if they are supplied as a list of character vectors rather than an "AAbin" object, in which case the k-mer length should be reduced (k < 4) to avoid excessive memory use and computation time.

## Value

Returns an object of class `"dendrogram"`.

## Author(s)

Shaun Wilkinson

## References

Edgar RC (2004) Local homology recognition and distance measures in linear time using compressed amino acid alphabets. *Nucleic Acids Research*, **32**, 380-385.

Paradis E, Claude J, Strimmer K, (2004) APE: analyses of phylogenetics and evolution in R language. *Bioinformatics* **20**, 289-290.

Paradis E (2012) Analysis of Phylogenetics and Evolution with R (Second Edition). Springer, New York.

## See Also

kcount

## Examples

```
## Not run:
## Cluster the woodmouse dataset (ape package)
library(ape)
data(woodmouse)
## trim gappy ends to subset global alignment
woodmouse <- woodmouse[, apply(woodmouse, 2, function(v) !any(v == 0xf0))]
## build tree divisively
set.seed(999)
woodmouse.tree <- topdown(woodmouse, nstart = 20)
## plot tree
op <- par(no.readonly = TRUE)
par(mar = c(5, 2, 4, 8) + 0.1)
plot(woodmouse.tree, main = "Woodmouse phylogeny", horiz = TRUE)
par(op)

## End(Not run)
```

## ultrametricize *Make dendrogram ultrametric.*

### Description

This is a simple function that sets the 'height' attributes of all leaf nodes to zero to aid vizualization.

### Usage

```
ultrametricize(x)
```

### Arguments

| | |
|---|---|
| x | an object of class "dendrogram". |

### Value

Returns an object of class "dendrogram".

### Author(s)

Shaun Wilkinson

### Examples

```
x <- read.dendrogram(text = "(A:0.1,B:0.2,(C:0.3,D:0.4):0.5);")
plot(x, horiz = TRUE)
x <- ultrametricize(x)
plot(x, horiz = TRUE)
```

## write.dendrogram *Export a dendrogram object to text.*

### Description

This function writes a dendrogram object to Newick-style parenthetic text.

### Usage

```
write.dendrogram(x, file = "", append = FALSE, edges = TRUE, ...)
```

## Arguments

| | |
|---|---|
| x | an object of class "dendrogram". |
| file | a character string naming a file or connection to write the output to. If no file path is specified or file = "" the result is printed to the console. |
| append | logical indicating whether the output should be appended to the file. If append = FALSE the contents of the file will be overwritten (the default setting). |
| edges | logical indicating whether edge weights should be included in the output string. |
| ... | further arguments to be passed to format. Used to specify the numbering style of the edge weights (if edges = TRUE). |

## See Also

read.dendrogram to create a "dendrogram" object from a text file. The write.tree function in the ape package performs a similar operation for "phylo" and "multiPhylo" objects.

## Examples

```
## build and export tree for the woodmouse data (ape package)
library(ape)
data(woodmouse)
## trim gappy ends for global alignment
woodmouse <- woodmouse[, apply(woodmouse, 2, function(v) !any(v == 0xf0))]
## build topdown tree
set.seed(999)
x <- topdown(woodmouse, nstart = 20)
write.dendrogram(x, edges = TRUE)
```

# Index