Package 'paleoDiv'

March 1, 2024

Title Extracting and Visualizing Paleobiodiversity

Version 0.0.6.0

Description Functions for conveniently downloading and editing taxon-specific datasets from the Paleobiology Database <https://paleobiodb.org>, extracting information on abundance, temporal distribution of subtaxa and taxonomic diversity through deep time, and visualizing these data in relation to phylogeny and stratigraphy.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0), strap, divDyn, ggplot2, knitr, rmarkdown

Config/testthat/edition 3

Depends R (>= 2.10), ape, stringr

LazyData true

VignetteBuilder knitr

NeedsCompilation no

Author Darius Nau [aut, cre]

Maintainer Darius Nau <dariusnau@gmx.at>

R topics documented:

`ab.gg` *Make a data.frame() that can be used to plot diversity data with density plots, e.g. in ggplot2*

Description

Make a data.frame() that can be used to plot diversity data with density plots, e.g. in ggplot2

Usage

```
ab.gg(data, taxa = NULL, agerange = c(252, 66), precision_ma = 1)
```

Arguments

`data`	list()-object containing occurrence data.frames or single occurrence data.frame()
`taxa`	Selection of taxa to include. If NULL, then abundance is tabulated for each unique factor level of data$tna
`agerange`	Range of geological ages to include in data.frame()
`precision_ma`	Size of intervals (in ma) at which to calculate diversity within the age range.

Details

Each taxon receives one entry per occurrence per time interval. The number of entries per taxon at any given point is thus proportional to the abundance of the taxon in the fossil record, and can be used for plotting with frequency- or density-based functions (e.g. hist(), ggplot2::geom_violin(), etc.). Note that using age values in the original occurrence table instead of this function will often be fully sufficient if the number of occurrences is considered an adequate proxy for abundance. However, instead using the ab.gg() and thus visualizing the results of the abdistr_() function has the benefit of the ability to account for a column of abundance values within the occurrence dataset, if available.

Value

A data.frame() with two columns: ma, for the numerical age, and tax, for the taxon.

Examples

```
data(archosauria)
ab.gg(archosauria, taxa=c("Pterosauria","Aves"), agerange=c(252,0),precision_ma=1)->flyers
library(ggplot2)
ggplot(data=flyers, aes(x=tax, y=ma))+ylim(252,0)+geom_violin(scale="count")
ggplot(data=flyers, aes(col=tax, x=ma))+xlim(252,0)+geom_density(adjust=0.5)
```

---

abdistr_ *Count number of entries in occurrence or collection data.frame for specific points in geological time*

Description

Count number of entries in occurrence or collection data.frame for specific points in geological time

Usage

```
abdistr_(
  x,
  table = NULL,
  ab.val = table$abund_value,
  smooth = 0,
  max = table$eag,
  min = table$lag,
  w = rep(1, length(x))
)
```

Arguments

x	A numeric vector giving the times (in ma) at which to determine the number of overlapping records.
table	An occurrence or collection dataset

ab.val	Abundance value to be used. Default is table$abund_value. If NULL (e.g. because this column doesn't exist) or NA, each occurrence is treated as representing one specimen
smooth	The smoothing margin, in units of ma. Corresponds to the plusminus parameter of rmeana(). Defaults to 0, i.e. no smoothing (beyond the resolution determined by the resolution of x)
max	Vector or column containing maximum age of each occurrence or collection
min	Vector or column containing minimum age of each occurrence or collection
w	A Vector of weights. Must be of same length as x

Value

A numeric vector of the same length as x, giving the estimated number of occurrence records (if ab.val==FALSE) or specimens (if ab.val==TRUE), or the estimated number of collections (if collection data are used instead of occurrences) overlapping each temporal value given in x

Examples

```
pdb("Stegosauria")->Stegosauria
abdistr_(x=c(170:120), table=Stegosauria)
```

---

add.alpha *Add transparency to any color*

Description

Add transparency to any color

Usage

```
add.alpha(col, alpha = 0.5)
```

Arguments

col	Color value or vector of colors
alpha	Opacity value to apply to the color(s)

Value

A character vector containing color hex codes.

Examples

```
add.alpha("red",0.8)
```

ages_archosauria *ages_archosauria*

Description

A dataset containing earliest and latest occurrence dates for clades shown in the example phylogeny.

Usage

```
ages_archosauria
```

Format

A matrix with 13 rows and 2 collumns containing:

FAD Earliest occurrence age

LAD Latest occurrence age

. . . for each taxon

archosauria *archosauria*

Description

A dataset of stratigraphic ranges of species within the clades in tree_archosauria.

Usage

```
archosauria
```

Format

A list() object containing 15 species tables (data.frames) with the following data in each:

tna taxon names (species names)

max maximum ages

min minimum ages

ma mean ages

Source

Generated from data downloaded from the paleobiology database https://paleobiodb.org using the functions pdb(), occ.cleanup() and mk.sptab()

`convert.sptab` *Convert geological ages in taxon-range tables as constructed by mk.sptab() for plotting alongside a time-calibrated phylogeny.*

Description

Convert geological ages in taxon-range tables as constructed by mk.sptab() for plotting alongside a time-calibrated phylogeny.

Usage

```
convert.sptab(sptab, tree = NULL, root.time = tree$root.time)
```

Arguments

`sptab`	Taxon-range table to convert
`tree`	Optional phylogenetic tree to draw root.time from
`root.time`	Root time of the tree, used for converting ages

Value

A data.frame() object in the format of the original taxon-range table, but with geological ages converted for plotting alongside the the phylogenetic tree.

Examples

```
data(archosauria)
data(tree_archosauria)
convert.sptab(archosauria$Coelophysoidea,tree_archosauria)
```

`darken` *Darken or lighten colors by adding/subtracting to or hsv channel values*

Description

Darken or lighten colors by adding/subtracting to or hsv channel values

Usage

```
darken(x, add = 0, abs = NULL)
```

Arguments

x	Color value or vector of colors
add	Value to be added to the third hsv-channel. Can be a vector of length x, or a vector of any length if length(x)==1
abs	Value to substitute for the third hsv-channel. If set, this overrides the setting for parameter add. Can be a vector of length x, or a vector of any length if length(x)==1

Value

A color value or vector of colour values of length x (or, if length(x)==1, the length of add or abs)

Examples

```
darken(ggcol(3),abs=0.5)
```

---

div.gg *Make a data.frame() that can be used to plot diversity data with density plots, e.g. in ggplot2*

Description

Make a data.frame() that can be used to plot diversity data with density plots, e.g. in ggplot2

Usage

```
div.gg(data, taxa, agerange = c(252, 66), precision_ma = 1, prefix = "sptab_")
```

Arguments

data	list()-object containing taxon-range tables
taxa	Selection of taxa to include
agerange	Range of geological ages to include in data.frame()
precision_ma	Size of intervals (in ma) at which to calculate diversity within the age range.
prefix	Prefix under which to find taxon-range tables in data

Details

Each taxon receives one entry per subtaxon (e.g. species) occurring for each time interval at which it occurs. The number of entries per taxon at any given point is thus proportional to the diversity of the taxon, and can be used to trick density functions (e.g. hist(), density()) into plotting diversity diagrams of various types. This is most useful when using ggplot2's geom_violin(), geom_histogram() or geom_density() functions. A simpler alternative to achieve a similar result would be to use the taxon-range-tables directly with these functions. However, this will lead to a relative underestimate of diversity for taxa with long-lived subtaxa, since each subtaxon will only be counted once. The div.gg()-function circumvents this problem by representing each taxon for each time interval in which it occurs, i.e. the relative number of entries in the returned data.frame will be proportional to the relative number of taxa with ranges overlapping each point in time.

Value

A data.frame() with two columns: ma, for the numerical age, and tax, for the taxon.

Examples

```
data(archosauria)
div.gg(archosauria, taxa=c("Pterosauria","Aves"), agerange=c(252,0),precision_ma=1)->flyers
library(ggplot2)
ggplot(data=flyers, aes(x=tax, y=ma))+ylim(252,0)+geom_violin(scale="count")
ggplot(data=flyers, aes(col=tax, x=ma))+xlim(252,0)+geom_density(adjust=0.5)
```

---

divdistr_ *Calculate total species diversity for any point in time based on a taxon-range table*

Description

Calculate total species diversity for any point in time based on a taxon-range table

Usage

```
divdistr_(
  x,
  table = NULL,
  w = rep(1, length(x)),
  smooth = 0,
  max = table$max,
  min = table$min
)
```

Arguments

x	A point in time or vector of points in time, in ma, at which species diversity is to be determined.
table	A taxon-range table to be used, usually the output of mk.sptab()
w	A vector of weights to apply to the estimated (raw) diversity figures. This vector needs to be of the same length as x. Each raw diversity estimate will then be multiplied by the weight. Can be used to account for differences in collection intensity/sampling biases, if these can be quantified (e.g. by analyzing collection records.
smooth	The smoothing margin, in units of ma. Corresponds to the plusminus parameter of rmeana(). Defaults to 0, i.e. no smoothing (beyond the resolution determined by the resolution of x)
max	Vector or column containing the maximum age of each entry in the taxon-range table. Defaults to table$max
min	Vector or column containing the minimum age of each entry in the taxon-range table. Defaults to table$min

Details

divdistr_() produces a "maximum" estimate of taxonomic diversity at any given point in time in the fossil record. This function is based on the principle of counting the number of taxon ranges (from the privided range table) that overlap each age provided in x. As a result of uncertainty of age estimates, this may lead to an overestimation of the actual fossil diversity at each point in time, especially at the points of overlap between taxon-specific ranges. Moreover this represents a "raw", uncorrected diversity estimate that does not account for differences in sampling intensity throughout the time interval that is investigated. A rudimentary functionality for using such a correction exists in the form of the w argument, which allows the user to provide a vector of weights (of the same length as x) to be multiplied with the raw diversity estimates. Such weights can, for instance, be based on (the inverse of) the number of collections overlapping any given age in x, which can be calculated using the same basic approach as the raw diversity, by downloading collections instead of occurrence data.

Value

A numeric vector containing taxon diversity (at the chosen taxonomic level used in the generation of the range table) at the provided ages.

Examples

```
data(archosauria)
divdistr_(c(170:140),table=archosauria$sptab_Stegosauria)
curve(divdistr_(x,archosauria$sptab_Stegosauria), xlim=c(200,100),ylim=c(-5,35))
ts.stages(ylim=c(-6,-1),alpha=0.3,border=add.alpha("grey"))
ts.periods(ylim=c(-6,-1),alpha=0.0)
```

---

divdistr_int *Count number of taxon records overlapping a specific time interval.*

Description

Count number of taxon records overlapping a specific time interval.

Usage

```
divdistr_int(x, table = NULL, ids = F, max = table$max, min = table$min)
```

Arguments

x	A numeric vector of length 2 specifying the start and end (in ma) of the time interval in question.
table	Taxon-range table to use
ids	Logical whether to return ids of entries in taxon-range table (defaults to FALSE) or their number
max	Vector or column containing the maximum age of each entry in the taxon-range table. Defaults to table$max

`min` Vector or column containing the minimum age of each entry in the taxon-range table. Defaults to table$min

Value

A single numeric giving the number of entries in table overlapping the specified interval, or a numeric vector giving their indices.

Examples

```
data(archosauria)
divdistr_int(x=c(201,220), table=archosauria$sptab_Coelophysoidea)
```

---

diversity_table *diversity_table*

Description

A dataset of diversity by stage, exemplifying the output produced by the divDyn-package.

Usage

```
diversity_table
```

Format

A data.frame() containing mean ages and diversity figures by stage.

x_orig ages for each stage in the phanerozoic

x ages converted for plotting on tree_archosauria, using the tsconv()-function

Sauroporomorpha diversity by stage for Sauropodomorpha

etc diversity by stage for each of the taxa represented in tree_archosauria ...

---

ggcol *Replicate the standard colour scheme from ggplot2*

Description

Replicate the standard colour scheme from ggplot2

Usage

```
ggcol(n)
```

Arguments

n	Length of colour vector to return.

Value

A character vector containing color hex codes.

Examples

```
ggcol(3)
```

---

mk.sptab *Generate a taxon-range table based on an occurrence dataset.*

Description

Generate a taxon-range table based on an occurrence dataset.

Usage

```
mk.sptab(
  xx = NULL,
  taxa = xx$tna,
  earliest = xx$eag,
  latest = xx$lag,
  tax = NULL
)
```

Arguments

xx	A data.frame() of occurrence records, containing at least the following columns: taxonomic name at level at which ranges are to be determined (e.g. species or genus), earliest possible age for each occurrence and latest possible age for each occurrence. If xx==NULL, then each column or vector must be specified individualy using the following parameters
taxa	column/vector containing the taxonomic variable. Defaults to xx$tna
earliest	column/vector containing the earliest age estimate. Defaults to xx$eag.
latest	column/vector containing the latest age estimate. Defaults to xx$lag.
tax	Optional. A single character string containing the taxon name, to be added as another column to the range table (useful for categorization, should several range tables be concatenated, e.g. using rbind()).

Value

A data.frame() containing the taxon names, the maximum and minimum age for each taxon, and (optionally) a column with the name of the higher-level taxon.

Examples

```
pdb("Stegosauria")->Stegosauria
mk.sptab(Stegosauria)->sptab_Stegosauria
```

`occ.cleanup` *Clean up occurrence dataset by removing commonly used character combinations in the identified name that will result in different factor levels for the same taxon.*

Description

Clean up occurrence dataset by removing commonly used character combinations in the identified name that will result in different factor levels for the same taxon.

Usage

```
occ.cleanup(x, remove = NULL)
```

Arguments

`x`	A occurrence data.frame or character vector containing the variable to clean up (defaults to x$tna)
`remove`	Which values to remove. If NULL, a default set of commonly occurring character combinations is used ("n. gen.", "n. sp.", "cf.","aff.", punctuation, as well as double, leading and ending spaces). If user-defined, remove needs to be formatted as a character vector with the values to be removed as names, i.e. in the format of c("remove_this" = "", "removethistoo"="")

Value

A character vector containing the cleaned up taxonomic names.

Examples

```
pdb("Coelophysoidea",full=TRUE)->coelo
occ.cleanup(coelo)->coelo$tna
```

pdb *Download data from the paleobiology database.*

Description

Download data from the paleobiology database.

Usage

```
pdb(taxon, interval = "all", what = "occs", full = F)
```

Arguments

taxon	A taxon (base_name) for which to download records.
interval	A character string indicating over which temporal interval to download data (defaults to "all"), e.g. "Phanerozoic" or "Jurassic".
what	The type of data to download (for details, see https://paleobiodb.org/data1.2/). Defaults to "occs", which downloads occurrence data. Setting this parameter to "colls" will instead download collection data.
full	A logical indicating whether or not the full dataset is to be downloaded (defaults to FALSE). At the expense of larger file size, the full dataset contains a large number of additional columns containing data such as stratigraphy, phylogeny and (paleo)geography, which is useful for various purposes but not strictly necessary for graphing paleodiversity.

Value

A data.frame() containing the downloaded paleobioDB dataset. The column "identified_name" will be copied into the column "tna", and (if what==occs) the columns "max_ma" and "min_ma" will be copied into the columns named "eag" and "lag" respectively, maintaining compatibility with the output of the deprecated package "paleobioDB" for those variable names.

Examples

```
pdb("Stegosauria")->Stegosauria
```

pdb.autodiv *A wrapper around pdb(), occ.cleanup() and mk.sptab() to automatically download and clean occurrence data from the paleobiology database and build species-level taxon-range tables for multiple taxa in one step.*

Description

A wrapper around pdb(), occ.cleanup() and mk.sptab() to automatically download and clean occurrence data from the paleobiology database and build species-level taxon-range tables for multiple taxa in one step.

Usage

```
pdb.autodiv(taxa, cleanup = TRUE, interval = NULL)
```

Arguments

taxa	Either a character vector of valid taxonomic names, or an object of class "phylo" whose tip.labels to use instead.
cleanup	Logical indicating whether to apply occ.cleanup() to occurrence data after download (defaults to TRUE)
interval	Stratigraphic interval for which to download data (defaults to NULL, which downloads data for all intervals)

Value

A list() object containing occurrence data (saved under the taxon names given) and species-level taxon-range tables (saved with the prefix "sptab_" before the taxon names).

Examples

```
pdb.autodiv("Coelophysoidea")->coelo
```

---

pdb.diff — *Subtract one occurrence data.frame from another, for disentangling overlapping taxonomies or quantifying stem-lineage diversity.*

Description

Subtract one occurrence data.frame from another, for disentangling overlapping taxonomies or quantifying stem-lineage diversity.

Usage

```
pdb.diff(x, subtract, id_col = x$occurrence_no)
```

Arguments

x	Occurrence data from which to subtract.
subtract	Occurrence data frame or vector of occurrence numbers to subtract from x
id_col	Vector or column of x containing id to be used for determining which values are also found in subtract or subtract$occurrence_no

Value

A data.frame() containing the difference between the two occurrence datasets, i.e. all entries that are in x but not in subtract.

Examples

```
pdb("Stegosauria")->Stegosauria
pdb("Thyreophora")->Thyreophora
pdb.diff(Thyreophora, subtract=Stegosauria)->non_stegosaur_thyreophorans
```

`pdb.union` *Form the union of two occurrence data.frames or remove duplicates from occurrence data.frame. Useful if parts of a clade are not included in the downloaded dataset and need to be added separately.*

Description

Form the union of two occurrence data.frames or remove duplicates from occurrence data.frame. Useful if parts of a clade are not included in the downloaded dataset and need to be added separately.

Usage

```
pdb.union(x, id_col = x$occurrence_no)
```

Arguments

`x`	Concatenated occurrence data.frames to be merged
`id_col`	Vector or column of x containing id to be used for determining which values contain occurrence numbers to be used for matching entries

Value

A data.frame() containing the first entry for each unique occurrence to be represented in x.

Examples

```
pdb("Stegosauria")->Stegosauria
pdb("Ankylosauria")->Ankylosauria
pdb.union(rbind(Ankylosauria, Stegosauria))->Eurypoda
```

`phylo.spindles` *Plots a phylogenetic tree with spindle-diagrams, optimized for showing taxonomic diversity.*

Description

Plots a phylogenetic tree with spindle-diagrams, optimized for showing taxonomic diversity.

Usage

```
phylo.spindles(
  phylo0,
  occ,
  stat = divdistr_,
  prefix = "sptab_",
  ages = NULL,
  xlimits = c(round(phylo0$root.time) - 1, 0),
  res = 1,
  weights = 1,
  dscale = 0.002,
  col = add.alpha("black"),
  fill = col,
  lwd = 1,
  lty = 1,
  cex.txt = 1,
  col.txt = add.alpha(col, 1),
  axis = T,
  labels = T,
  txt.y = 0.5,
  txt.x = mean(xlimits),
  add = FALSE,
  tbmar = 0.2,
  smooth = 0
)
```

Arguments

`phylo0`	A time-calibrated phylogenetic tree to plot
`occ`	Either a list()-object containing taxon-range tables for plotting diversity, or a matrix() or data.frame()-object that contains numerical plotting statistics. If the latter is provided, the default use of divdistr_() is overridden and the function will look for a column named "x" and columns matching the phylogeny tip.labels to plot the spindles.
`stat`	Plotting statistic to be passed on to viol(). Defaults to use divdistr_().
`prefix`	Prefix for taxon-range tables in occ. Defaults to "sptab_"

ages	Optional matrix with lower and upper age limits for each spindle, formatted like the output of tree.ages() (most commonly the same calibration matrix used to time-calibrate the tree)
xlimits	Limits for plotting the phylogeny on the x axis.
res	Temporal resolution of diversity estimation
weights	Weights for diversity estimation. Must have the same length as the range of xlimits divided by res. For details, see divdistr_()
dscale	Scale value of the spindles on the y axis. Should be adjusted manually to optimize visibility of results.
col	Color to use for the border of the plotted spindles
fill	Color to use for the fill of the plotted spindles. Defaults to col.
lwd	Line width for the plotted spindles.
lty	Line type for the plotted spindles.
cex.txt	Adjustment for tip label text size
col.txt	tip label text color, defaults to be same as col, but with no transparency
axis	Logical indicating whether to plot (temporal) x axis (defaults to TRUE)
labels	Logical indicating whether to plot tip labels of phylogeny (defaults to TRUE)
txt.y	y axis alignment of tip labels
txt.x	x coordinates for plotting tip labels. Can be a single value applicable to all labels, or a vector of the same length as phylo0$tip.label
add	Logical indicating whether to add to an existing plot, in which case only the spindles are plotted on top of an existing phylogeny, or not, in which case the phylogeny is plotted along with the spindles.
tbmar	Top and bottom margin around the plot. Numeric of either length 1 or 2
smooth	Smoothing parameter to be passed on to divdistr_()

Details

The phylo.spindles() function allows the plotting of a phylogeny with spindle diagrams at each of its terminal branches. Various data can be represented (e.g. disparity, abundance, various diversity measures, such as those output by the divDyn package, etc.) depending on the settings for occ and stat, but the function is optimized to plot the results of divdistr_() and does do by default. If another function is used as an argument to stat, it has to be able to take occ as its argument and return a vector of the same length as range(xlimits)/res to be plotted on the phylogeny. If occ is a list() object containing multiple dataframes, occurrence datasets of taxon range tables are automatically converted to work with abdistr_() or divdistr_() respectively.

Examples

```
data(archosauria)
data(tree_archosauria)
data(ages_archosauria)
data(diversity_table)
phylo.spindles(tree_archosauria,occ=archosauria,dscale=0.005,ages=ages_archosauria,txt.x=66)
phylo.spindles(tree_archosauria,occ=diversity_table,dscale=0.005,ages=ages_archosauria,txt.x
```

`redraw.phylo` *Redraw the lines of a phylogenetic tree.*

Description

Redraw the lines of a phylogenetic tree.

Usage

```
redraw.phylo(
  saved_plot = NULL,
  col = "black",
  lwd = 1,
  lty = 1,
  lend = 2,
  arrow.l = 0,
  arrow.angle = 45,
  arrow.code = 2,
  indices = NULL
)
```

Arguments

`saved_plot`	Optional saved plot (e.g. using get("last_plot.phylo", envir = ape::.PlotPhyloEnv)) to be used instead of currently active plot.
`col`	Color to be used for redrawing tree edges.
`lwd`	Line width to be used for redrawing tree edges.
`lty`	Line type to be used for redrawing tree edges.
`lend`	Style of line ends to be used for redrawing tree edges.
`arrow.l`	Length of arrow ends to be used for plotting. Defaults to 0, i.e. no visible arrow.
`arrow.angle`	Angle of arrow ends to be used for plotting. Defaults to 45°.
`arrow.code`	Arrow code to be used for plotting. For details, see ?arrows
`indices`	Optional indices which edges to redraw. Can be used to highlight specific edges in different color or style.

Value

Plots a timescale on the currently active plot.

Examples

```
data(tree_archosauria)
ape::plot.phylo(tree_archosauria)
redraw.phylo(col="darkred",lwd=3,indices=c(19:24))
redraw.phylo(col="red",lwd=3,indices=c(18),arrow.l=0.1)
```

rmean — *Calculate a rolling mean for a vector x.*

Description

Calculate a rolling mean for a vector x.

Usage

```
rmean(x, width = 11)
```

Arguments

x	Numeric vector for which to calculate the rolling mean.
width	Width of the interval over which to calculate rolling mean values. Should be an uneven number (even numbers are coerced into the next-higher uneven number)

Value

A numeric vector of the same length as x containing the calculated rolling means, with the first and last few values being NA (depending on the setting for width)

Examples

```
rmean(x=c(1,2,3,4,5,6),width=5)
```

rmeana — *Calculate a rolling mean based on distance within a second variable.*

Description

Calculate a rolling mean based on distance within a second variable.

Usage

```
rmeana(x0, y0, x1 = NULL, plusminus = 5, weighting = FALSE, weightdiff = 0)
```

Arguments

x0	Numeric independent variable at which rolling mean is to be calculated.
y0	Numeric variable of which mean is to be calculated.
x1	Optional. New x values at which rolling mean of y0 is to be calculated. If x1==NULL, calculation will take place at original (x0) values.

plusminus	Criterium for the width (in x0) of the interval over which rolling mean values are to be calculated. Value represents the margin as calculated from every value of x1 or x0, i.e. for a plusminus==5, the interval over which the means are drawn will range from values with x-x_i=5 to x-x_i=-5.
weighting	Whether or not to apply weighting. If weighting==TRUE, then means are calculated as weighted means with weighting decreasing linearly towards the margins of the interval over which the mean is to be drawn.
weightdiff	Minimum weight to be added to all weights if weighting==TRUE. Defaults to 0.

Value

A numeric vector of the same length as either x1 (if not NULL) or x0, containing the calculated rolling means.

Examples

```
rmeana(x0=c(1,2,3,4,5,6), y0=c(2,3,3,4,5,6))
```

---

stax.sel *Extract subsets of an occurrence data.frame.*

Description

Extract subsets of an occurrence data.frame.

Usage

```
stax.sel(taxa, rank = x$class, x = NULL)
```

Arguments

taxa	A vector containing subtaxa (or any other entries matching entries of rank) to be returned
rank	Vector or column of x in which to look for entries matching taxa. defaults to x$class, for selecting class-level subtaxa from large datasets (only works if pdb(...,full=TRUE)
x	Optional occurrence data.frame. If set, a data.frame with the selected entries will be returned.

Value

If is.null(x) (default), a vector giving the indices of values matching taxa in rank. Otherwise, an occurrence data.frame() containing only the selected taxa or values.

Examples

```
pdb("Coelophysoidea",full=TRUE)->coelo
stax.sel(c("Coelophysis"), rank=coelo$genus,x=coelo)->Coelophysis
stax.sel(c("Carnian","Norian","Rhaetian"), rank=coelo$early_interval)->triassic_coelophysoid
```

`tree.ages` *Automatically build matrix for time-calibration of phylogenetic trees using occurrence data*

Description

Automatically build matrix for time-calibration of phylogenetic trees using occurrence data

Usage

```
tree.ages(phylo0 = NULL, data = NULL, taxa = phylo0$tip.label)
```

Arguments

`phylo0`	(Optional) Object of class=="phylo" from which to draw taxa to include in calibration matrix
`data`	Optional list()-object containing either taxon-range tables or occurrence datasets for all taxa. If NULL, data will be automatically downloaded via the pdb()-function
`taxa`	Taxa to include in calibration matrix, defaults to phylo0$tip.label

Value

A two-column matrix containing earliest and latest occurrences for each taxon in taxa, with taxon names as row names

Examples

```
data(archosauria)
data(tree_archosauria)
tree.ages(tree_archosauria,data=archosauria)->ages
```

`tree_archosauria` *tree_archosauria*

Description

A time-calibrated phylogenetic tree of Archosauria.

Usage

```
tree_archosauria
```

Format

An object of class==phylo with 13 tips and 12 internal nodes.

`ts.periods` *Add a horizontal, period-level phanerozoic timescale to any plot, especially calibrated phylogenies plotted with ape.*

Description

Add a horizontal, period-level phanerozoic timescale to any plot, especially calibrated phylogenies plotted with ape.

Usage

```
ts.periods(
  phylo = NULL,
  alpha = 1,
  names = T,
  exclude = c("Quarternary"),
  col.txt = NULL,
  border = NA,
  ylim = 0.5,
  adj.txt = c(0.5, 0.5),
  txt.y = mean,
  bw = F,
  update = NULL
)
```

Arguments

phylo	Optional (calibrated) phylogeny to which to add timescale. If phylogeny is provided, the $root.time variable is used to convert ages so that the time scale will fit the phylogeny.
alpha	Opacity value to use for the fill of the time scale
names	Logical indicating whether to plot period names (defaults to TRUE)
exclude	Character vector listing periods for which to not plot the names, if names==TRUE
col.txt	Color(s) to use for labels.
border	Color to use for the border of the timescale
ylim	Setting for height of the timescale. Can either be one single value, in which case the function attempts to use the lower limit of the currently plotted phylogeny, or a vector of length 2 containing the lower and upper limits of the timescale.
adj.txt	Numeric vector of length==2 giving horizontal and vertical label alignment (defaults to centered, i.e. 0.5 for both values)
txt.y	Function to use to determine the vertical text position (defaults to mean, i.e. centered)
bw	Logical whether to plot in black and white (defaults to FALSE). If TRUE, time scale is drawn with a white background
update	Character string giving the filename of a .csv table for providing an updated timescale. If provided, the values for plotting the time scale are taken from the csv file instead of the internally provided values. Table must have columns named periods, bottom, top and col, giving the period names, start time in ma, end time in ma and a valid color value, respectively.

Value

Plots a timescale on the currently active plot.

Examples

```
data(tree_archosauria)
ape::plot.phylo(tree_archosauria)
ts.periods(tree_archosauria, alpha=0.5)
```

---

ts.stages *Add a horizontal, stage-level phanerozoic timescale to any plot, especially calibrated phylogenies plotted with ape.*

Description

Add a horizontal, stage-level phanerozoic timescale to any plot, especially calibrated phylogenies plotted with ape.

Usage

```
ts.stages(
  phylo = NULL,
  alpha = 1,
  names = F,
  col.txt = NULL,
  border = NA,
  ylim = 0.5,
  adj.txt = c(0.5, 0.5),
  txt.y = mean,
  bw = F,
  update = NULL
)
```

Arguments

phylo	Optional (calibrated) phylogeny to which to add timescale. If phylogeny is provided, the $root.time variable is used to convert ages so that the time scale will fit the phylogeny.
alpha	Opacity value to use for the fill of the time scale
names	Logical indicating whether to plot stage names (defaults to FALSE)
col.txt	Color(s) to use for labels.
border	Color to use for the border of the timescale
ylim	Setting for height of the timescale. Can either be one single value, in which case the function attempts to use the lower limit of the currently plotted phylogeny, or a vector of length 2 containing the lower and upper limits of the timescale.
adj.txt	Numeric vector of length==2 giving horizontal and vertical label alignment (defaults to centered, i.e. 0.5 for both values)
txt.y	Function to use to determine the vertical text position (defaults to mean, i.e. centered)
bw	Logical whether to plot in black and white (defaults to FALSE). If TRUE, time scale is drawn with a white background
update	Character string giving the filename of a .csv table for providing an updated timescale. If provided, the values for plotting the time scale are taken from the csv file instead of the internally provided values. Table must have columns named stage, bottom, top and col, giving the stage names, start time in ma, end time in ma and a valid color value, respectively.

Value

Plots a timescale on the currently active plot.

Examples

```
data(tree_archosauria)
ape::plot.phylo(tree_archosauria)
```

```
ts.stages(tree_archosauria, alpha=0.7)
ts.periods(tree_archosauria, alpha=0)
```

`tsconv` *Convert geological ages for accurate plotting alongside a calibrated phylogeny*

Description

Convert geological ages for accurate plotting alongside a calibrated phylogeny

Usage

```
tsconv(x, phylo0 = NULL, root.time = phylo0$root.time)
```

Arguments

`x`	A vector of geological ages to be converted.
`phylo0`	Phylogeny from which to take root.age
`root.time`	Numeric root age, if not taken from a phylogeny

Value

A numeric() containing the converted geological ages

Examples

```
tsconv(c(252,201,66), root.time=300)
```

`viol` *Generate a violin plot*

Description

Generate a violin plot

Usage

```
viol(
  x,
  pos,
  x2 = NULL,
  stat = density,
  dscale = 1,
  cutoff = range(x),
  horiz = TRUE,
  add = T,
  lim = cutoff,
  xlab = "",
  ylab = "",
  fill = "grey",
  col = "black",
  lwd = 1,
  lty = 1,
  ...
)
```

Arguments

x	Variable for which to plot violin.
pos	Position at which to place violin in the axis perpendicular to x
x2	Optional variable to use instead of x as input variable for the violin plot. If x2 is set, the function (default: density()) used to calculate the plotting statistic is run on x2 instead of x, but the results are plotted at the corresponding x values.
stat	The plotting statistic. Details to the density() function, as in a standard violin plot, but can be overridden with another function that can take x or x2 as its first argument. Stat can also be a numeric vector of the same length as x, in which case the values in this vectors are used instead of the function output and plotted against x as an independent variable.
dscale	The scale to apply to the values for density (or another plotting statistic). Defaults to 1, but adjustment may be needed depending on the scale of the plot the violin is to be added to.
cutoff	Setting for cropping the violin. Can be either a single value, in which case the input is interpreted as number of standard deviations from the mean, or a numeric vector of length 2, giving the lower and upper cutoff value directly.
horiz	Logical indicating whether to plot horizontally (defaults to TRUE) or vertically
add	Logical indicating whether to add to an existing plot (defaults to TRUE) or generate a new plot.
lim	Limits (in the dimensions of x) used for plotting, if add==FALSE. Defaults to cutoff, but can be manually set as a numeric vector of length 2, giving the lower and upper limits of the plot.
xlab	x axis label

ylab	y axis label
fill	Fill color for the plotted violin
col	Line color for the plotted violin
lwd	Line width for the plotted violin
lty	Line width for the plotted violin
...	Other arguments to be passed on to function in parameter stat

Details

Viol provides a versatile function for generating violin plots and adding them to r base graphics. The default plotting statistic is density(), resulting in the standard violin plot. However, density can be overridden by entering any function that can take x or x2 as its first argument, or any numeric vector containing the data to be plotted, as long as this vector is the same length as x.

Value

A violin plot and a data.frame containing the original and modified plotting statistic and independent variable against which it is plotted.

Examples

```
viol(x=c(1,2,2,2,3,4,4,3,2,2,3,3,4,5,3,3,2,2,1,6,7,6,9),pos=1, add=FALSE)
viol(c(1:10), width=9, stat=rmean, pos=0, add=FALSE)
viol(c(1:10), stat=c(11:20), pos=0, add=FALSE)
```

Index