Package 'evolved'

November 27, 2024

Title Open Software for Teaching Evolutionary Biology at Multiple Scales Through Virtual Inquiries

Version 1.0.0

Description ``Evolutionary Virtual Education'' - 'evolved' - provides multiple tools to help educators (especially at the graduate level or in advanced undergraduate level courses) apply inquiry-based learning in general evolution classes. In particular, the tools provided include functions that simulate evolutionary processes (e.g., genetic drift, natural selection within a single locus) or concepts (e.g. Hardy-Weinberg equilibrium, phylogenetic distribution of traits). More than only simulating, the package also provides tools for students to analyze (e.g., measuring, testing, visualizing) datasets with characteristics that are common to many fields related to evolutionary biology. Importantly, the package is heavily oriented towards providing tools for inquiry-based learning - where students follow scientific practices to actively construct knowledge. For additional details, see package's vignettes.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Imports ape, diversitree, phytools

Suggests knitr, rmarkdown, testthat (>= 3.0.0), LSD, paleobioDB, BAMMtools, plot3D

Config/testthat/edition 3

Depends R (>= 2.10)

LazyData true

VignetteBuilder knitr

URL <https://github.com/Auler-J/evolved>

BugReports https://github.com/Auler-J/evolved/issues

NeedsCompilation no

Author Matheus Januario [aut, cph, cre]
(<https://orcid.org/0000-0002-6480-7095>),
Jennifer Auler [aut] (<https://orcid.org/0000-0001-7576-9058>),
Andressa Viol [aut],
Daniel Rabosky [aut]

Maintainer Matheus Januario <januarioml.eco@gmail.com>

Repository CRAN

Date/Publication 2024-11-27 12:50:02 UTC

Contents

`birds_spp` *Birds species list*

Description

Birds species list

Usage

```
birds_spp
```

Format

A `vector` containing the names of almost all (i.e., 9993) extant bird species, following Jetz et al. (2012) taxonomy. This dataset is part of the package and is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0).

Source

Actual file downloaded from https://vertlife.org/data/

References

Jetz, W., Thomas, G. H., Joy, J. B., Hartmann, K., & Mooers, A. O. (2012). The global diversity of birds in space and time. Nature, 491(7424), 444-448.

calcFossilDivTT *Calculate paleo diversity curves through different methods*

Description

`calcFossilDivTT` calculates fossil diversity through time using different methods.

Usage

```
calcFossilDivTT(
  data,
  tax.lvl = "species",
  method = "rangethrough",
  bin.reso = 1
)
```

Arguments

`data` A `data.frame` containing the columns: `max_ma`, `min_ma` and the name provided in `tax.lvl`. `max_ma` and `min_ma` are respectively the early and late bounds of rock layer's age. `tax.lvl` column is the taxonomic level of the data. Any additional columns are ignored.

`tax.lvl` A `character` giving the taxonomic in which calculations will be based on (default value is `"species"`). This must refer to the column names in `data`.

`method` A `character` string setting the method which should be used. Could be either `"rangethrough"` or `"stdmethod"`, which will respectively calculate diversity using the range through or the standard methods (Foote & Miller, 2007)

`bin.reso` A `numeric` assigning the resolution (length) of the time bin to consider in calculations. Default value is 1 (which in most cases - e.g. those following the Paleobiology Database default timescale - will equate to one million years)

Value

A `data.frame` containing the diversity (column `div`) of the chosen taxonomic level through time, with calculation based on `method`. If `"method = rangethrough"`, the time moments are the layer boundaries given in `data`. If `"method = stdmethod"`, the time moments are evenly-space bins with length equal to `bin.reso`, starting at the earliest bound in the dataset.

Author(s)

Matheus Januario, Jennifer Auler

References

Foote, M., Miller, A. I., Raup, D. M., & Stanley, S. M. (2007). Principles of paleontology. Macmillan.

Examples

```
# Loading data
data("dinos_fossil")

# Using function:
div1 <- calcFossilDivTT(dinos_fossil, method = "stdmethod")
div2 <- calcFossilDivTT(dinos_fossil, method = "stdmethod", bin.reso = 10)

# Comparing different bins sizes in the standard method
plot(x=div1$age, y=div1$div, type="l",
     xlab = "Time (Mya)", ylab = "Richness",
     xlim=rev(range(div1$age)), col="red")
lines(x=div2$age, y=div2$div, col="blue")

# Comparing different methods:
div3 <- calcFossilDivTT(dinos_fossil, method = "rangethrough")
plot(x=div1$age, y=div1$div, type="l",
     xlab = "Time (Mya)", ylab = "Richness",
     xlim=rev(range(div1$age)), col="red")
lines(x=div3$age, y=div3$div, col="blue")
```

checkAndFixUltrametric

Find and fix small rounding errors in ultrametric trees

Description

checkAndFixUltrametric finds and correct small numerical errors that might appear in ultrametric trees that where created through simulations. This function should never be used as a formal statistical method to make a tree ultrametric, as it was designed just to correct small rounding errors.

Usage

```
checkAndFixUltrametric(phy)
```

Arguments

phy	A phylo object, following terminology from package ape in which function will operate.

Value

A check and fixed phylo object.

Author(s)

Daniel Rabosky, Matheus Januario, Jennifer Auler

References

Paradis, E. (2012). Analysis of Phylogenetics and Evolution with R (Vol. 2). New York: Springer.

Popescu, A. A., Huber, K. T., & Paradis, E. (2012). ape 3.0: New tools for distance-based phylogenetics and evolutionary analysis in R. Bioinformatics, 28(11), 1536-1537.

Examples

```
S <- 1
E <- 0
set.seed(1)
phy <- simulateTree(pars = c(S, E), max.taxa = 6, max.t = 5)
phy$edge.length[1] <- phy$edge.length[1]+0.1
ape::is.ultrametric(phy)
phy <- checkAndFixUltrametric(phy)
ape::is.ultrametric(phy)
```

countSeqDiffs *Counting protein sequence differences*

Description

countSeqDiffs counts the number of protein differences among two sequences of proteins within the same "ProteinSeq" object.

Usage

```
countSeqDiffs(x, taxon1, taxon2)
```

Arguments

x	A "ProteinSeq" object containing proteins from taxon1 and taxon2.
taxon1	A character giving the common name of the first species that will be compared. Must be a name present in x.
taxon2	A character giving the common name of the second species that will be compared. Must be a name present in x.

Value

A integer giving the number of protein differences between taxon1 and taxon2.

Author(s)

Matheus Januario, Dan Rabosky, Jennifer Auler

Examples

```
countSeqDiffs(cytOxidase, "human", "chimpanzee")

countSeqDiffs(cytOxidase, "human", "cnidaria")

countSeqDiffs(cytOxidase, "chimpanzee", "cnidaria")
```

---

cytOxidase *Cytochrome Oxidase sequences*

Description

`cytOxidase` is a set of homologous protein sequences from the GENE cytochrome oxidase SUBUNIT 1 gene. This mitochondrial gene, often known as CO1 (“see-oh-one”), plays a key role in cellular respiration. C01 contains approximately 513 aminoacids and has been used by previous studies for reconstructing phylogenetic trees and estimating divergence times in Metazoaria by assuming a molecular clock. Its 5’ partition is used for the ‘Barcoding of Life’ initiative, for instance. This dataset is part of the package and is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0).

Usage

```
cytOxidase
```

Format

A object of class "ProteinSeq" with 17 entries, each representing a different animal species

names Organism’s popular name or a taxonomic classification

sequence Aminoacid sequence of length 513

Details

The object of class "ProteinSeq" is structured as a named `vector` of 17 different animal species, with each individual component being a sequence of 513 aminoacids.

Source

Amino acid sequences were originally downloaded from genebank and later curated and aligned by Daniel L. Rabosky.

`data_whales` *Whale body size and speciation rates*

Description

Data on the body size of many cetacean species and species-specific speciation rates. This dataset is part of the package and is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0).

Usage

```
data_whales
```

Format

A `data.frame` with 75 rows and 4 columns.

species Whale species

log_mass Log of body mass (grams)

S Species-specific speciation rate

color Suggested color to be used for the tip's clade

Details

Species follow taxonomy from Steeman et al (2009). Species-specific speciation rates from Rabosky 2014 & Rabosky et al, 2014. Mass data from PanTHERIA (Jones et al, 2009).

Source

Compilation of many primary sources (see details).

References

Jones, K. E., Bielby, J., Cardillo, M., Fritz, S. A., O'Dell, J., Orme, C. D. L., ... & Purvis, A. (2009). PanTHERIA: a species-level database of life history, ecology, and geography of extant and recently extinct mammals: Ecological Archives E090-184. Ecology, 90(9), 2648-2648.

Rabosky, D. L. (2014). Automatic detection of key innovations, rate shifts, and diversity-dependence on phylogenetic trees. PLoS one, 9(2), e89543.

Rabosky, D. L., Grundler, M., Anderson, C., Title, P., Shi, J. J., Brown, J. W., ... & Larson, J. G. (2014). BAMM tools: an R package for the analysis of evolutionary dynamics on phylogenetic trees. Methods in Ecology and Evolution, 5(7), 701-707.

Steeman, M. E., Hebsgaard, M. B., Fordyce, R. E., Ho, S. Y., Rabosky, D. L., Nielsen, R., ... & Willerslev, E. (2009). Radiation of extant cetaceans driven by restructuring of the oceans. Systematic biology, 58(6), 573-585.

`dinos_fossil` *Occurrence of dinosaur fossils*

Description

Many dinosaur (including avian species) fossil occurrences from different moments of the geological past. Much information (i.e., extra columns) was removed from the original dataset to make it more compact, but it can be fully accessed by the data URL. This dataset is part of the package and is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0).

Usage

```
dinos_fossil
```

Format

A `data.frame` containing 15527 rows and 13 columns

phylum Organism phylum

class Organism taxonomic class

order Organism taxonomic order

family Organism taxonomic family

genus Organism genus

species Organism specific name

early_interval Earlier known geological period of occurrence

late_interval Later known geological period of occurrence

max_ma Occurrence's oldest time boundary in million years

min_ma Occurrence's newest time boundary in million years

midpoint Midpoint between max_ma and min_ma

lng Longitude of place where occurrence was found. Follows decimal degree format.

lat Latitude of place where occurrence was found. Follows decimal degree format.

Source

The Paleobiology Database (downloaded on 2022-03-11).
Data URL: http://paleobiodb.org/data1.2/occs/list.csv?datainfo&rowcount&base_name=Dinosauria&show=full,classext,gen

estimateSpeciation *Estimate speciation assuming a pure-birth process*

Description

estimateSpeciation Estimates the speciation rate assuming a constant-rate, pure-birth model.

Usage

```
estimateSpeciation(phy)
```

Arguments

phy	A phylo object, following terminology from package ape in which function will operate.

Value

A numeric with the estimated speciation rate.

Author(s)

Daniel Rabosky, Matheus Januario, Jennifer Auler

References

Yule G.U. 1925. A mathematical theory of evolution, based on the conclusions of Dr. JC Willis, FRS. Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of a Biological Character. 213:21–87.

Examples

```
S <- 1
E <- 0
set.seed(1)
phy <- simulateTree(pars = c(S, E), max.taxa = 6, max.t = 5)
estimateSpeciation(phy)
```

fitCRBD *Fit a constant-rate birth-death process to a phylogeny*

Description

fitCRBD fits a constant-rate birth-death process to a phylogeny in the format of ape package's phylo object. Optimization is based on likelihood functions made with diversitree. This function is basically a wrapper for the diversitree's make.bd function.

Usage

```
fitCRBD(phy, n.opt = 5, l.min = 0.001, l.max = 5, max.bad = 200)
```

Arguments

phy	A phylo object, following terminology from package ape, in which function will operate.
n.opt	Number of optimizations that will be tried by function.
l.min	Lower bound for optimization. Default value is 0.001.
l.max	Upper bound for optimization. Default value is 5.
max.bad	Maximum number of unsuccessful optimization attempts. Default value is 200.

Value

A numeric with the best estimates of speciation S and extinction E rates.

Author(s)

Daniel Rabosky, Matheus Januario, Jennifer Auler

References

Paradis, E. (2012). Analysis of Phylogenetics and Evolution with R (Vol. 2). New York: Springer.

Popescu, A. A., Huber, K. T., & Paradis, E. (2012). ape 3.0: New tools for distance-based phylogenetics and evolutionary analysis in R. Bioinformatics, 28(11), 1536-1537.

FitzJohn, R. G. (2010). Analysing diversification with diversitree. R Package. ver, 9-2.

FitzJohn, R. G. (2012). Diversitree: comparative phylogenetic analyses of diversification in R. Methods in Ecology and Evolution, 3(6), 1084-1092.

See Also

see help page from diversitree::make.bd and stats::optim

Examples

```
S <- 0.1
E <- 0.1
set.seed(1)
phy <- simulateTree(pars = c(S, E), max.taxa = 30, max.t = 8)
fitCRBD(phy)
```

`lttPlot` *Make a lineage through time (LTT) plot*

Description

`lttPlot` plots the lineage through time (LTT) of a `phylo` object. It also adds a reference line connecting the edges of the graph.

Usage

```
lttPlot(
  phy,
  lwd = 1,
  col = "red",
  plot = TRUE,
  rel.time = FALSE,
  add = FALSE,
  knitr = FALSE
)
```

Arguments

`phy` A `phylo` object, as specified by the `ape` package.

`lwd` Line width.

`col` Line color.

`plot` A `logical` indicating with calculations should be plotted. If `FALSE`, function returns a list of the calculated points.

`rel.time` A `logical` indicating how the time scale should be shown. If `FALSE` (default), plots the absolute time since `phy`'s crown age. If `TRUE`, plots time as a relative proportion between crown age and furthest tip from root.

`add` A `logical` indicating if plot should be added to pre-existing plot. Default is `FALSE`.

`knitr` Logical indicating if plot is intended to show up in RMarkdown files made by the `Knitr` R package.

Value

Plots the sum of alive lineages per point in time, and adds a red line as a reference of expectation under pure birth. If `plot = FALSE`, a list the richness of each point in time, and `phy`'s crown age.

Author(s)

Daniel Rabosky, Matheus Januario, Jennifer Auler

References

Paradis, E. (2012). Analysis of Phylogenetics and Evolution with R (Vol. 2). New York: Springer.

Examples

```
S <- 1
E <- 0
set.seed(1)
phy <- simulateTree(pars = c(S, E), max.taxa = 20, max.t = 5)
lttPlot(phy, knitr = TRUE)
lttPlot(phy, plot = FALSE, knitr = TRUE)
```

`NatSelSim` *Simulating natural selection through time in a bi-allelic gene*

Description

`NatSelSim` simulates natural selection in a bi-allelic gene through `n.gen` generations.

Usage

```
NatSelSim(
  w11 = 1,
  w12 = 1,
  w22 = 0.9,
  p0 = 0.5,
  n.gen = 10,
  plot.type = "animateall",
  print.data = FALSE,
  knitr = FALSE
)
```

Arguments

`w11` Number giving the fitness of genotype A1A1. Values will be normalized if any genotype fitness exceeds one.

`w12` Number giving the fitness of genotype A1A2. Values will be normalized if any genotype fitness exceeds one.

`w22` Number giving the fitness of genotype A2A2. Values will be normalized if any genotype fitness exceeds one.

`p0` Initial (time = 0) allelic frequency of A1. A2's initial allelic frequency is `1-p0`.

`n.gen` Number of generation that will be simulated.

`plot.type` String indicating if plot should be animated. The default, "animateall" animate all possible panels. Other options are "static" (no animation), "animate1", "animate3", or "animate4". Users can animate each panel individually (using `plot.type = "animateX"`, with X being the panel which one wants to animate (so options are "animate1", "animate3", and "animate4" (see return for more info).

`print.data` Logical indicating whether all simulation results should be returned as a `data.frame`. Default value is `FALSE`.

`knitr` Logical indicating if plot is intended to show up in RMarkdown files made by the `Knitr` R package.

Details

If any value of fitness (i.e., `w11`, `w12`, `w22`) is larger than one, fitness is interpreted as absolute fitness and values are re-normalized.

Value

If `print.data = TRUE`, it returns a `data.frame` containing the number of individuals for each genotype through time. The plots done by the function shows (1) Allele frequency change through time. (2) The adaptive landscape (which remains static during the whole simulation, so can't be animated), (3) Time series of mean population fitness, and (4) Time series of genotypic population frequencies.

Author(s)

Matheus Januario, Jennifer Auler, Dan Rabosky

References

Fisher, R. A. (1930). The Fundamental Theorem of Natural Selection. In: The genetical theory of natural selection. The Clarendon Press

Plutynski, A. (2006). What was Fisher's fundamental theorem of natural selection and what was it for?. Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences, 37(1), 59-82.

Examples

```
#using the default values (w11=1, w12=1, w22=0.9, p0=0.5, n.gen=10)
NatSelSim()

# Continuing a simulation for extra time:
# Run the first simulation
sim1=NatSelSim(w11 = .4, w12 = .5, w22 = .4, p0 = 0.35,
n.gen = 5, plot.type = "static", print.data = TRUE, knitr = TRUE)

# Then take the allelic frequency form the first sim:
new_p0 <- (sim1$AA[nrow(sim1)] + sim1$Aa[nrow(sim1)]*1/2)
```

```
# and use as p0 for a second one:

NatSelSim(w11 = .4, w12 = .5, w22 = .4, p0 = new_p0, n.gen = 5, plot.type = ”static”, knitr = TRUE)
```

`OneGenHWSim` *Simulating one generation of genotypes under Hardy-Weinberg equilibrium*

Description

`OneGenHWSim` creates `n.sim` simulations of one generation of genotypes under Hardy-Weinberg equilibrium for a bi-allelic loci.

Usage

```
OneGenHWSim(n.ind = 50, p = 0.5, n.sim = 100)
```

Arguments

`n.ind` Integer indicating the census size of the simulated populations. If decimals are inserted, they will be rounded.

`p` Numerical between zero and one that indicates A1’s allele frequency. A2’s allele frequency is assumed to be `1-p`.

`n.sim` Number of simulations to be made. If decimals are inserted, they will be rounded.

Value

A `data.frame` containing the number of individuals for each genotype.

Author(s)

Matheus Januario, Dan Rabosky, Jennifer Auler

References

Hardy, G. H. (1908). Mendelian proportions in a mixed population. Science, 28, 49–50.

Weinberg, W. (1908). Uber den Nachweis der Vererbung beim Menschen. Jahreshefte des Vereins fur vaterlandische Naturkunde in Wurttemberg, Stuttgart 64:369–382. [On the demonstration of inheritance in humans]. Translation by R. A. Jameson printed in D. L. Jameson (Ed.), (1977). Benchmark papers in genetics, Volume 8: Evolutionary genetics (pp. 115–125). Stroudsburg, PA: Dowden, Hutchinson & Ross.

Mayo, O. (2008). A century of Hardy–Weinberg equilibrium. Twin Research and Human Genetics, 11(3), 249-256.

Examples

```
#using the default values (n.ind = 50, p = 0.5, n.sim = 100):
OneGenHWSim()

#Simulating with a already fixed allele:
OneGenHWSim(n.ind = 50, p = 1)

# Testing if the simulation works:
A1freq <- .789 #any value could work
n.simul <- 100
simulations <- OneGenHWSim(n.ind = n.simul, n.sim = n.simul, p = A1freq)

#expected:
c(A1freq^2, 2*A1freq*(1-A1freq), (1-A1freq)^2)

#simulated:
apply(X = simulations, MARGIN = 2, FUN = function(x){mean(x)/n.simul})
```

`plotNatSel` *Plot NatSelSim output*

Description

Plot NatSelSim output

Usage

```
plotNatSel(
  gen.HW = gen.HW,
  p.t = p.t,
  w.t = w.t,
  t = t,
  W.gntp = c(w11, w12, w22),
  plot.type = "animateall",
  knitr = FALSE
)
```

Arguments

`gen.HW` Dataframe with A1A1, A1A2 and A2A2 genotypic frequencies in each generation (nrows = NGen)

`p.t` Allelic frequency through time

`w.t` Mean population fitness through time

`t` time

`W.gntp` Initial genotypic fitness

`plot.type`	String indicating if plot should be animated. The default, "animateall", animate all possible panels. Other options are "static", "animate1", "animate3", or "animate4".
`knitr`	Logical indicating if plot is intended to show up in RMarkdown files made by the `Knitr` R package.

Value

Plot of NatSelSim's output (see `NatSelSim`'s help page for details).

`plotPaintedWhales` *Plotting the whale phylogeny and coloring its clades*

Description

`plotPaintedWhales` plots the phylogeny from Steeman et al (2011), coloring the Dolphins (Delphinidae), porpoises (Phocoenidae), the Mysticetes, the baleen whales (Balaenopteridae), and the Beaked whales (Ziphiidae).

Usage

```
plotPaintedWhales(
  show.legend = TRUE,
  direction = "rightwards",
  knitr = FALSE,
  ...
)
```

Arguments

`show.legend`	Logical indicating if clade legend should be shown.
`direction`	Phylogeny plotting direction. Should be set to "rightwards"
`knitr`	Logical indicating if plot is intended to show up in RMarkdown files made by the `Knitr` R package. (the default) or "leftwards".
`...`	other arguments to be passed to `phytools::plotSimmap`

Value

The whale phylogeny, with branch lengths being colored by a major whale taxonomic group.

Author(s)

Matheus Januario, Jennifer Auler

References

Steeman, M. E., Hebsgaard, M. B., Fordyce, R. E., Ho, S. Y., Rabosky, D. L., Nielsen, R., ... & Willerslev, E. (2009). Radiation of extant cetaceans driven by restructuring of the oceans. Systematic biology, 58(6), 573-585.

See Also

help page from `phytools::plotSimmap`

Examples

```
plotPaintedWhales(knitr = TRUE)
```

`plotProteinSeq` *Plot protein sequence(s)*

Description

`plotProteinSeq` draws the sequences of proteins within the same "ProteinSeq" object. In this format, more similar sequences will have similar banding patterns.

Usage

```
plotProteinSeq(x, taxon.to.plot, knitr = FALSE)
```

Arguments

`x`	A "ProteinSeq" object containing proteins from `taxon1` and `taxon2`.
`taxon.to.plot`	A character vector providing the common name of the species that will be plotted. Must be a name present in `x`.
`knitr`	Logical indicating if plot is intended to show up in RMarkdown files made by the `Knitr` R package.

Value

A draw of the protein sequence(s) provided. Colors refer to specific amino acids ("R", "W", "I", "F", "S", "T", "N", "H", "K", "D", "G", "L", "Y", "V", "M", "A", "E", "P", "Q", "C")", "gaps/space in the sequence ("-"), ambiguous amino acid ("B" - often representing either asparagine ("N") or aspartic acid ("D")), or another marker for ambiguous amino acid ("X").

Author(s)

Matheus Januario, Jennifer Auler

Examples

```
data(cytOxidase)
plotProteinSeq(cytOxidase, c("human", "chimpanzee", "cnidaria"), knitr = TRUE)
```

plotRawFossilOccs *Plot a literal interpretation of a fossil record*

Description

plotRawFossilOccs calculates and plots the early and late boundaries associated with each taxa in a dataset.

Usage

```
plotRawFossilOccs(
  data,
  tax.lvl = NULL,
  sort = TRUE,
  use.midpoint = TRUE,
  return.ranges = FALSE,
  knitr = FALSE
)
```

Arguments

data
: A data.frame containing fossil data on the age (early and late bounds of rock layer, respectively labeled as max_ma and min_ma) and the taxonomic level asked in tax_lv.

tax.lvl
: A character giving the taxonomic in which calculations will be based on, which must refer to the column names in data. If NULL (default value), the function plots every individual occurrences in data.

sort
: logical indicating if taxa should be sorted by their max_ma values (default value is TRUE). Otherwise (i.e., if FALSE), function will follow the order of taxa (or occurrences) inputted in data.

use.midpoint
: logical indicating if function should use occurrence midpoints (between max_ma and min_ma) as occurrence temporal boundaries, a method commonly employed in paleobiology to remove noise related to extremely coarse temporal resolution due to stratification. This argument is only used if a tax.lvl is provided.

return.ranges
: logical indicating if ranges calculated by function should be return as a data.frame. If tax.lvl is NULL, the function don't calculate ranges and so it has nothing to return.

knitr
: Logical indicating if plot is intended to show up in RMarkdown files made by the Knitr R package.

Value

Plots a pile of the max-min temporal ranges of the chosen tax.lvl. This usually will be stratigraphic ranges for occurrences (so there is no attempt to estimate "true" ranges), and if tax.lvl = NULL (the default), occurrences are drawn as ranges of stratigraphic resolution (= the fossil dating imprecision). If return.ranges = TRUE, it returns a data.frame containing the diversity (column div) of the chosen taxonomic level, through time.

Author(s)

Matheus Januario, Jennifer Auler

Examples

```
data("dinos_fossil")
oldpar <- par(no.readonly = TRUE)
par(mfrow=c(1,2))
plotRawFossilOccs(dinos_fossil, tax.lvl = "species", knitr = TRUE)
plotRawFossilOccs(dinos_fossil, tax.lvl = "genus", knitr = TRUE)
par(oldpar)
```

plotWFDrift *Plot WFDriftSim output*

Description

Plot WFDriftSim output

Usage

```
plotWFDrift(p.through.time, plot.type = plot, knitr = FALSE)
```

Arguments

p.through.time	Matrix with n.gen columns and n.sim lines
plot.type	String. Options are "static" or "animate"
knitr	Logical indicating if plot is intended to show up in RMarkdown files made by the Knitr R package.

Value

A static or animated plot of populations under genetic drift through time

Examples

```
store_p = WFDriftSim(Ne = 5, n.gen = 10, p0=.2, n.sim=5, plot = "none", print.data = TRUE)
plotWFDrift(store_p, "static")
```

ProteinSeq *Details, generics, and methods for the* ProteinSeq *class*

Description

The ProteinSeq class is an input for the functions countSeqDiffs and is.ProteinSeq. It consists of a character vector. Each entry in this vector represents the aminoacid (the protein components coded by a gene) sequence, for a given aligned protein sequence. The object must be a character, named vector, with the names typically corresponding to the species (name could be scientific or common name) from which every sequence came. The characters within the vector must correspond to valid aminoacid symbols (i.e. capitalized letters or deletion "_" symbols). Particularly, the following symbols relate to amino acids: "A", "C", "D", "E", "F", "G", "H", "I", "K", "L", "M", "N", "P", "Q", "R", "S", "T", "V", "W", "Y".

Importantly, the symbol "_" means an indel (insertion or deletion), and the symbols "X", "B", "Z", "J" should be considered as ambiguous site readings.

Usage

```
is.ProteinSeq(x)

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

## S3 method for class 'ProteinSeq'
summary(object, ...)

## S3 method for class 'ProteinSeq'
head(x, n = 20, ...)

## S3 method for class 'ProteinSeq'
tail(x, n = 20, ...)
```

Arguments

x	an object of the class ProteinSeq
...	arguments to be passed to or from other methods. #' @return Shows the last n elements of a ProteinSeq object.
object	an object of the class ProteinSeq
n	number of aminoacids to be shown

Details

is.ProteinSeq A ProteinSeq must be a list containing multiple vectors made of characters (usually letters that code to Amino Acids, deletions, etc). All of these must have the correct length (i.e. same as all the others) and their relative positions should match (i.e., the object must contain *alligned* Amino acide sequences).

Value

A logical indicating if object `x` is of class `ProteinSeq`

`print.ProteinSeq` Prints a brief summary of a `print.ProteinSeq` containing the number of sequences and the length of the alignment. See more details of the format in `??ProteinSeq`.

Same as `print.ProteinSeq`.

Shows the first `n` elements of a `ProteinSeq` object.

Author(s)

Daniel Rabosky, Matheus Januario, Jennifer Auler

`simulateBirthDeathRich` *Simulating richness through birth-death processes*

Description

`simulateBirthDeathRich` calculates the number of species at a certain point in time, following a birth-death process.

Usage

```
simulateBirthDeathRich(t, S = NULL, E = NULL, K = NULL, R = NULL)
```

Arguments

`t`	Point in time which richness will be simulated.
`S`	A numeric representing the per-capita speciation rate (in number of events per lineage per million years). Must be larger than `E`.
`E`	A numeric representing the per-capita extinction rate (in number of events per lineage per million years). Must be smaller than `S`.
`K`	A numeric representing the extinction fraction (i.e., `K = E / S`). Must be either zero or a positive which is number smaller than one.
`R`	A numeric representing the per-capita Net Diversification rate (i.e., `R = S - E`). Must be a positive number.

Details

The function only accepts as inputs `S` and `E`, or `K` and `R`.

Value

The number of simulated species (i.e., the richness).

Author(s)

Matheus Januario, Daniel Rabosky, Jennifer Auler

References

Raup, D. M. (1985). Mathematical models of cladogenesis. Paleobiology, 11(1), 42-52.

Examples

```
# running a single simulation:
SS <- 0.40
EE <- 0.09
tt <- 10 #in Mya
simulateBirthDeathRich(t = tt, S = SS, E = EE)

#running many simulations and graphing results:
nSim <- 1000
res <- vector()
for(i in 1:nSim){
  res <- c(res,
  simulateBirthDeathRich(t = tt, S = SS, E = EE))
}
plot(table(res)/length(res),
     xlab="Richness", ylab="Probability")
```

---

simulateTree *Simulating a phylogenetic trees through the birth-death process*

Description

`simulateTree` uses a birth-death process to simulate a phylogenetic tree, following the format of ape package's `phylo` object. The function is basically a wrapper for the `diversitree`'s `tree.bd` function.

Usage

```
simulateTree(
  pars,
  max.taxa = Inf,
  max.t,
  min.taxa = 2,
  include.extinct = FALSE
)
```

Arguments

`pars`	`numeric` vector with the simulation parameters: speciation (first slot) and extinction (second slot) rates, respectively. Should follow any formats stated in the function `tree.bd` from the `diversitree` package.
`max.taxa`	Maximum number of taxa to include in the tree. If `Inf`, then the tree will be evolved until `max.t` time has passed.
`max.t`	Maximum length to evolve the phylogeny over. If equal to `Inf`, then the tree will evolve until `max.taxa` extant taxa are present.
`min.taxa`	Minimum number of taxa to include in the tree.
`include.extinct`	A `logical` indicating if extinct taxa should be included in the final phylogeny.

Details

see help page from `diversitree::tree.bd`

Value

A `phylo` object

Author(s)

Daniel Rabosky, Matheus Januario, Jennifer Auler

References

Paradis, E. (2012). Analysis of Phylogenetics and Evolution with R (Vol. 2). New York: Springer.

Popescu, A. A., Huber, K. T., & Paradis, E. (2012). ape 3.0: New tools for distance-based phylogenetics and evolutionary analysis in R. Bioinformatics, 28(11), 1536-1537.

FitzJohn, R. G. (2010). Analysing diversification with diversitree. R Packag. ver, 9-2.

FitzJohn, R. G. (2012). Diversitree: comparative phylogenetic analyses of diversification in R. Methods in Ecology and Evolution, 3(6), 1084-1092.

Examples

```
S <- 1
E <- 0
set.seed(1)
phy <- simulateTree(pars = c(S, E), max.taxa = 6, max.t=Inf)
ape::plot.phylo(phy)
ape::axisPhylo()

# alternatively, we can stop the simulation using time:
set.seed(42)
phy2 <- simulateTree(pars = c(S, E), max.t=7)
ape::plot.phylo(phy2)
ape::axisPhylo()
```

timeseries_fossil *Fossil Time series*

Description

Values of clade diversity for many clades of organisms (note some clades are nested within other clades in the dataset). This dataset is part of the package and is licensed =under the Creative Commons Attribution 4.0 International License (CC BY 4.0).

Usage

```
timeseries_fossil
```

Format

A `data.frame` with 598 rows and 6 columns.

clade Time series clade

source Primary source of the Time series

stem_age Stem age of clade

rel_time Geological relative time (in Million years ago relative to present)

time_ma Geological time in million years since clade stem age

richness Number of species at given geological time

Details

Legend:
anth = Anthozoa (Cnidaria);
art = Articulata (Crinoidea, Echinodermata);
biv = Bivalvia (Mollusca);
bryo = Bryozoa (Lophotrochozoa, Ectoprocta);
ceph = Cephalopoda (Mollusca);
chon = Chondrocytes (Chordata);
crin = Crinoidea (Echinodermata);
dinosauria = Dinosauria (Chordata);
ech = Echinoidea (Echinodermata);
foram = Foraminifera (Retaria);
gast = Gastropoda (Mollusca);
graptoloids = Graptolites (Graptolithina);
ling = Ligulata (Brachiopoda);
ostr = Ostracoda (Crustacea, Arthropoda);
tril = Trilobita (Arthropoda).

Source

Data originally compiled from many primary sources. Organized, curated by, and downloaded from, Rabosky & Benson (2021).

References

Rabosky, D. L., & Benson, R. B. (2021). Ecological and biogeographic drivers of biodiversity cannot be resolved using clade age-richness data. Nature Communications, 12(1), 2945.

WFDriftSim — *Simulating generations of genetic drift in a Wright–Fisher (WF) population*

Description

`WFDriftSim` simulates genetic drift of diploid Wright–Fisher populations with a given effective population size through a certain number of generations.

Usage

```
WFDriftSim(
  Ne,
  n.gen,
  p0 = 0.5,
  n.sim = 1,
  plot.type = "animate",
  print.data = FALSE,
  knitr = FALSE
)
```

Arguments

`Ne`	Number giving the effective population size of the population
`n.gen`	Number of generations to be simulated.
`p0`	Initial frequency of a given allele. As the simulated organism is diploid, the other alleles frequency will be `1-(p0)`. Default value is `0.5`.
`n.sim`	Number of simulations to be made. If decimals are inserted, they will be rounded. Default value is `1`.
`plot.type`	Character indicating if simulations should be plotted as colored lines. Each color represents a different population. If `plot.type = "animate"` (default value) it animates each generation individually. If `plot.type = "static"` it plots all lines rapidly. If `plot.type = "none"` nothing is plotted.
`print.data`	Logical indicating whether all simulation results should be returned as a `data.frame`. Default value is `FALSE`.
`knitr`	Logical indicating if plot is intended to show up in RMarkdown files made by the `Knitr` R package.

Details

The effective population size (Ne) is strongly connected with the rate of genetic drift (for details, see Waples, 2022).

Value

If plot.type = "static" or "animate", plots the timeseries of all simulations, with each line+color referring to a different simulation. Note that if many simulations (generally more than 20) are simulated, colors might be cycled and different simulation will have the same color. If print.data = TRUE, returns a data.frame with the simulation results.

Author(s)

Matheus Januario, Dan Rabosky, Jennifer Auler

References

Fisher RA (1922) On the dominance ratio. Proc. R. Soc. Edinb 42:321–341

Kimura M (1955) Solution of a process of random genetic drift with a continuous model. PNAS–USA 41(3):144–150

Tran, T. D., Hofrichter, J., & Jost, J. (2013). An introduction to the mathematical structure of the Wright–Fisher model of population genetics. Theory in Biosciences, 132(2), 73-82. [good for the historical review, math can be challenging]

Waples, R. S. (2022). What is Ne, anyway?. Journal of Heredity.

Wright S (1931) Evolution in Mendelian populations. Genetics 16:97–159

Examples

```
#Default values:
WFDriftSim(Ne = 5, n.gen = 10, knitr = TRUE)

#A population which has already fixed one of the alleles:
WFDriftSim(Ne = 5, n.gen = 10, p0=1, knitr = TRUE)

#Many populations::
WFDriftSim(Ne = 5, n.gen = 10, p0=0.2, n.sim=10, knitr = TRUE)

######## continuing a previous simulation:
n.gen_1stsim <- 10 # number of gens in the 1st sim:
sim1 <- WFDriftSim(Ne = 5, n.gen = n.gen_1stsim, p0=.2, n.sim=10,
plot.type = "none", print.data = TRUE, knitr = TRUE)
n.gen_2ndsim <-7 # number of gens in the 2nd sim:
# now, note how we assigned p0:
sim2 <- WFDriftSim(Ne = 5, n.gen = n.gen_2ndsim, p0=sim1[,ncol(sim1)],
plot.type = "static", n.sim=10, print.data = TRUE, knitr = TRUE)

# if we want to merge both simulations, then we have to:
# remove first column of 2nd sim (because it repeats
# the last column of the 1st sim)
```

```
sim2 <- sim2[,-1]

# re-name 2nd sim columns:
colnames(sim2) <- paste0("gen", (n.gen_1stsim+1):(n.gen_1stsim+n.gen_2ndsim))

#finally, merging both rounds of simulations:
all_sims <- cbind(sim1, sim2)
head(all_sims)
```

whale_phylo *Whale Phylogeny*

Description

An ultrametric phylogenetic tree of the living cetaceans. Phylogeny generated by Steeman et al (2009). This dataset is part of the package and is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0).

Usage

```
whale_phylo
```

Format

An ultrametric phylo object with 87 tips

Source

Original phylogeny generation by Steeman et al (2001). File obtained from Rabosky et al, 2014.

References

Rabosky, D. L., Grundler, M., Anderson, C., Title, P., Shi, J. J., Brown, J. W., ... & Larson, J. G. (2014). BAMM tools: an R package for the analysis of evolutionary dynamics on phylogenetic trees. Methods in Ecology and Evolution, 5(7), 701-707.

Steeman, M. E., Hebsgaard, M. B., Fordyce, R. E., Ho, S. Y., Rabosky, D. L., Nielsen, R., ... & Willerslev, E. (2009). Radiation of extant cetaceans driven by restructuring of the oceans. Systematic biology, 58(6), 573-585.

Index