

Package ‘velociraptr’

February 15, 2017

Type Package

Title Fossil Analysis

Version 1.0

Author Andrew A. Zaffos

Maintainer Andrew A Zaffos <azaffos@wisc.edu>

Description Functions for downloading, reshaping, culling, cleaning, and analyzing fossil data from the Paleobiology Database <<https://paleobiodb.org>>.

License GPL-3

Encoding UTF-8

Depends rgdal,RCurl

LazyData true

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-15 20:25:34

R topics documented:

abundanceMatrix	2
adaptiveOrigination	3
ageRanges	4
cleanTaxonomy	5
constrainAges	6
cullMatrix	7
downloadPaleogeography	8
downloadPBDB	9
downloadTime	10
multiplicativeBeta	11
presenceMatrix	12
subsampleEvenness	13
subsampleIndividuals	14
taxonAlpha	15
uniformExtinction	17

abundanceMatrix	<i>Create a community matrix of taxon abundances.</i>
-----------------	---

Description

Creates a community matrix of taxon abundances, with samples as rows and species as columns, from a data frame.

Usage

```
abundanceMatrix(Data, Rows = "geoplate", Columns = "genus")
```

Arguments

Data	A data.frame of taxonomic occurrences. Must have at least two columns. One column representing the samples, and one column representing the taxa.
Rows	A character string
Columns	A character string

Value

A numeric matrix of taxon abundances. Samples as the rownames and species as the column names.

Author(s)

Andrew A. Zaffos

Examples

```
# Download a test dataset of pleistocene bivalves.  
# DataPBDB<-downloadPBDB(Taxa="Bivalvia", StartInterval="Pleistocene", StopInterval="Pleistocene")  
  
# Clean the genus column  
# DataPBDB<-cleanTaxonomy(DataPBDB,"genus")  
  
# Create a community matrix of genera by tectonic plate id#  
# CommunityMatrix<-abundanceMatrix(Data=DataPBDB, Rows="geoplate", Columns="genus")
```

adaptiveOrigination *Adaptive Beta Functions*

Description

Functions for placing confidence intervals on time of origination or extinction using the Adaptive Beta Method.

Usage

```
adaptiveOrigination(Ages, Confidence = 0.95)
```

```
adaptiveExtinction(Ages, Confidence = 0.95)
```

Arguments

Ages a numeric vector of fossil occurrence ages, expressed as millions of years ago.
Confidence the desired confidence level

Details

Takes a numeric vector of fossil occurrence ages, expressed as millions of years ago, to estimate either the time of origination `adaptiveOrigination` or extinction `adaptiveExtinction`. This code was originally presented in Wang et al. (2015) "Adaptive credible intervals on stratigraphic ranges when recovery potential is unknown". *Paleobiology* 42:240256. This simplified version will only accept ages in millions of years. The full version can be found in the online supplement to the paper.

Note that this function will not accept more than 161 age observations. This is a limitation of the original function.

Value

A matrix listing the oldest or youngest observed fossil occurrence, the best inferred age of extinction or origination, and the upper confidence limit.

Author(s)

Andrew A. Zaffos

Examples

```
# Generate an example dataset of fossil ages
FakeAges<-runif(50,50,100)

# Calculate the inferred age of extinction with 95% confidence.
adaptiveExtinction(FakeAges,Confidence=0.95)

# Calculate the inferred age of origination with 50% confidence.
```

```
adaptiveOrigination(FakeAges,Confidence=0.5)
```

ageRanges	<i>Find the age range for each taxon in a dataframe</i>
-----------	---

Description

Find the age range (first occurrence and last occurrence) for each taxon in a PBDB dataset. Can be run for any level of the taxonomic hierarchy (e.g., family, genus).

Usage

```
ageRanges(Data, Taxonomy = "genus")
```

Arguments

Data	A data frame downloaded from the paleobiology database API.
Taxonomy	A character string identifying the desired level of the taxonomic hierarchy.

Details

rangeDiversity converts the output of ageRanges into a vector of range-through richness in million year increments. Note that this function is hard-coded to the default field names from the paleobiology database.

Value

A numeric matrix of first and last ages for each taxon, with tax as rownames.

Author(s)

Andrew A. Zaffos

Examples

```
# Download a test dataset of Cenozoic bivalves.
# DataPBDB<-downloadPBDB(Taxa="Bivalvia",StartInterval="Cenozoic",StopInterval="Cenozoic")

# Find the first occurrence and last occurrence for all Cenozoic bivalves in DataPBDB
# AgeRanges<-ageRanges(DataPBDB,"genus")
```

cleanTaxonomy	<i>Clean taxonomic names</i>
---------------	------------------------------

Description

Removes NAs and subgenera from the genus column.

Usage

```
cleanTaxonomy(Data, Taxonomy = "genus")
```

Arguments

Data	A data frame of taxonomic occurrences downloaded from the paleobiology database API.
Taxonomy	A character string

Details

Will remove NA's and subgenera from the genus column of a PBDB dataset. It can also be used on other datasets of similar structure to convert species names to genus, or remove NAs.

Value

Will return a data frame identical to the original, but with the genus column cleaned.

Author(s)

Andrew A. Zaffos

Examples

```
# Download a test dataset of Cenozoic bivalves.  
# DataPBDB<-downloadPBDB(Taxa="Bivalvia",StartInterval="Cenozoic",StopInterval="Cenozoic")  
  
# Clean up the genus column.  
# CleanedPBDB<-cleanTaxonomy(DataPBDB,"genus")
```

constrainAges	<i>Constrain a dataset to only occurrences within a certain age-range</i>
---------------	---

Description

Assign fossil occurrences to different intervals within a geologic timescale, then remove occurrences that are not temporally constrained to a single interval within that timescale.

Usage

```
constrainAges(Data, Timescale)
```

```
multiplyAges(Data, Timescale)
```

Arguments

Data	A data frame
Timescale	A data frame

Details

Cull a paleobiology database data frame to only occurrences temporally constrained to be within a certain level of the geologic timescale (e.g., period, epoch). The geologic timescale should come from the Macrostrat database, but custom time-scales can be used if structured in the same way. See `downloadTime` for how to download a timescale.

Value

A data frame

Author(s)

Andrew A. Zaffos

Examples

```
# Download a test dataset of Cenozoic bivalves.
# DataPBDB<-downloadPBDB(Taxa="Bivalvia",StartInterval="Cenozoic",StopInterval="Cenozoic")

# Download the international epochs timescale from macrostrat.org
# Epochs<-downloadTime("international epochs")

# Find only occurrences that are temporally constrained to a single international epoch
# ConstrainedPBDB<-constrainAges(DataPBDB,Timescale=Epochs)

# Create multiple instances of a single occurrence for each epoch it occurs in
# MultipliedPBDB<-multiplyAges(DataPBDB,Timescale=Epochs)
```

cullMatrix	<i>Cull rare taxa and depauperate samples</i>
------------	---

Description

Functions for recursively culling community matrices of rare taxa and depauperate samples.

Usage

```
cullMatrix(CommunityMatrix, Rarity = 2, Richness = 2, Silent = FALSE)
```

Arguments

CommunityMatrix	a matrix
Rarity	a whole number
Richness	a whole number
Silent	logical

Details

Takes a community matrix (see `presenceMatrix` or `abundanceMatrix`) and removes all samples with fewer than a certain number of taxa and all taxa that occur below a certain threshold of samples. The function operates recursively, and will check to see if removing a rare taxon drops a sample below the input minimum richness and vice-versa. This means that it is possible to eliminate all taxa and samples if the rarity and richness minimums are too high. If the `Silent` argument is set to `FALSE` the function will throw an error and print a warning if no taxa or samples are left after culling. If `Silent` is set to `TRUE` the function will simply return `NULL`. The latter case is useful if many matrices are being culled as a part of a loop, and you do not want to break the loop with an error.

These functions originally appeared in the R script appendix of Holland, S.M. and A. Zaffos (2011) "Niche conservatism along an onshore-offshore gradient". *Paleobiology* 37:270-286.

Value

A community matrix with depauperate samples and rare taxa removed.

Author(s)

Andrew A. Zaffos

Examples

```
# Download a test dataset of pleistocene bivalves.
# DataPBDB<-downloadPBDB(Taxa="Bivalvia",StartInterval="Pleistocene",StopInterval="Pleistocene")

# Create a community matrix with tectonic plates as "samples".
# CommunityMatrix<-abundanceMatrix(DataPBDB,"geoplate")

# Remove taxa that occur in less than 5 samples and samples with fewer than 25 taxa.
# cullMatrix(CommunityMatrix,Rarity=5,Richness=25,Silent=FALSE)
```

downloadPaleogeography

Downloads paleogeographic maps

Description

Download a paleogeographic map for an age expressed in millions of years ago.

Usage

```
downloadPaleogeography(Age = 0)
```

Arguments

Age A whole number up to 550

Details

Downloads a map of paleocontinents for a specific age from Macrostrat.org as a shapefile. The given age must be expressed as a whole number. Note that the function makes use of the rgdal and RCurl packages.

Value

An rgdal compatible shapefile

Author(s)

Andrew A. Zaffos

Examples

```
# Download a test dataset of Maastrichtian bivalves.
# DataPBDB<-downloadPBDB(Taxa="Bivalvia",StartInterval="Maastrichtian",StopInterval="Maastrichtian")

# Download a paleogeographic map.
# KTBoundary<-downloadPaleogeography(Age=66)
```



```
# Plot the paleogeographic map (uses rgdal) and the PBDB points.
# plot(KTBoundary,col="grey")
# points(x=DataPBDB[, "paleolng"],y=DataPBDB[, "paleolat"],pch=16,cex=2)
```

downloadPBDB

Download Occurrences from the Paleobiology Database

Description

Downloads a data frame of Paleobiology Database fossil occurrences.

Usage

```
downloadPBDB(Taxa, StartInterval = "Pliocene", StopInterval = "Pleistocene")
```

Arguments

Taxa	a character vector
StartInterval	a character vector
StopInterval	a character vector

Details

Downloads a data frame of Paleobiology Database fossil occurrences matching certain taxonomic groups and age range. This is simply a convenience function for rapid data download, and only returns the most generically useful fields. Go directly to the Paleobiology Database to make more complex searches or access additional fields. This function makes use of the RCurl package.

- `occurrence_no`: The Paleobiology Database occurrence number.
- `collection_no`: The Paleobiology Database collection number.
- `reference_no`: The Paleobiology Database reference number.
- `Classifications`: The stated Linnean classification of the occurrence from phylum through genus. See `cleanTaxonomy` for how to simplify these fields.
- `accepted_name`: The highest resolution taxonomic name assigned to the occurrence.
- `Geologic Intervals`: The earliest possible age of the occurrence and latest possible age of the occurrence, expressed in terms of geologic intervals. See `constrainAge` for how to simplify these fields.
- `Numeric Ages`: The earliest possible age of the occurrence and latest possible age of the occurrence, expressed as millions of years ago.
- `Geolocation`: Both present-day and rotated paleocoordinates of the occurrence. The geoplate id used by the rotation model is also included. The key for geoplate ids can be found in the Paleobiology Database API documentation.

Value

a data frame

Author(s)

Andrew A. Zaffos

Examples

```
# Download a test dataset of Ypresian bivalves.
# DataPBDB<-downloadPBDB(Taxa="Bivalvia",StartInterval="Ypresian",StopInterval="Ypresian")

# Download a test dataset of Ordovician-Silurian trilobites and brachiopods.
# DataPBDB<-downloadPBDB(c("Trilobita","Brachiopoda"),"Ordovician","Silurian")
```

downloadTime

Download geologic timescale

Description

Downloads a geologic timescale from the Macrostrat.org database.

Usage

```
downloadTime(Timescale = "interational epochs")
```

Arguments

Timescale character string; a recognized timescale in the Macrostrat.org database

Details

Downloads a recognized timescale from the Macrostrat.org database. This includes the name, minimum age, maximum age, midpoint age, and official International Commission on Stratigraphy color hexcode if applicable of each interval in the timescale. Go to <https://macrostrat.org/api/defs/timescales?all> for a list of recognized timescales.

Value

A data frame

Author(s)

Andrew A. Zaffos

Examples

```
# Download the ICS recognized periods timescale
Timescale<-downloadTime(Timescale="international periods")
```

multiplicativeBeta *Multiplicative Diversity Partitioning*

Description

Calculates beta diversity under various Multiplicative Diversity Partitioning paradigms.

Usage

```
multiplicativeBeta(CommunityMatrix)
completeTurnovers(CommunityMatrix)
notEndemic(CommunityMatrix)
```

Arguments

CommunityMatrix
a matrix

Details

Takes a community matrix (see `presenceMatrix` or `abundanceMatrix`) and returns one of three types of multiplicative beta diversity discussed in Tuomisto, H (2010) "A diversity of beta diversities: straightening up a concept gone awry. Part 1. Defining beta diversity as a function of alpha and gamma diversity". *Ecography* 33:2-22.

- `multiplicativeBeta(CommunityMatrix)`: Calculates the original beta diversity ratio - $\text{Gamma}/\text{Alpha}$. It quantifies how many times as rich gamma is than alpha.
- `completeTurnovers(CommunityMatrix)`: The number of complete effective species turnovers observed among compositional units in the dataset - $(\text{Gamma}-\text{Alpha})/\text{Alpha}$.
- `notEndemic(CommunityMatrix)`: The proportion of taxa in the dataset not limited to a single sample - $(\text{Gamma}-\text{Alpha})/\text{Gamma}$

Value

A numeric vector

Author(s)

Andrew A. Zaffos

Examples

```
# Download a test dataset of pleistocene bivalves from the Paleobiology Database.
# DataPBDB<-downloadPBDB(Taxa="Bivalvia","Pleistocene","Pleistocene")

# Create a community matrix with tectonic plates as "samples".
# CommunityMatrix<-abundanceMatrix(DataPBDB,"geoplate")

# "True local diversity ratio"
# multiplicativeBeta(CommunityMatrix)

# Whittaker's effective species turnover
# completeTurnovers(CommunityMatrix)

# Proportional effective species turnover
# notEndemic(CommunityMatrix)
```

presenceMatrix *Create a matrix of presences and absences*

Description

Creates a community matrix of taxon presences and absences from a data frame with a column of sites and a column of species.

Usage

```
presenceMatrix(Data, Rows = "geoplate", Columns = "genus")
```

Arguments

Data	A dataframe or matrix
Rows	A character string
Columns	A character string

Value

A presence-absence matrix

Author(s)

Andrew A. Zaffos

Examples

```
# Download a test dataset of pleistocene bivalves.
# DataPBDB<-downloadPBDB(Taxa="Bivalvia", "Pleistocene", "Pleistocene")

# Create a community matrix of genera by plates.
# CommunityMatrix<-presenceMatrix(DataPBDB, Rows="geoplate", Columns="genus")

# Create a community matrix of families by geologic interval.
# CommunityMatrix<-presenceMatrix(DataPBDB, Rows="early_interval", Columns="family")
```

subsampleEvenness *Shareholder Quorum Subsampling*

Description

Calculate the richness of a sample after subsampling based on the evenness of the abundance distribution.

Usage

```
subsampleEvenness(Abundance, Quota = 0.9, Trials = 100,
  IgnoreSingletons = FALSE, ExcludeDominant = FALSE)
```

Arguments

Abundance	A vector of taxon abundances
Quota	A numeric value greater than zero and less than one
Trials	Number of iterations
IgnoreSingletons	A logical
ExcludeDominant	A logical

Details

This is a port of S.M. Holland's port of J. Alroy's shareholder quorum subsampling function.

Alroy, J. (2010) "Fair sampling of taxonomic richness and unbiased estimation of origination and extinction rates" *in* Quantitative Methods in Paleobiology, Paleontological Society Short Course 2010. The Paleontological Society Papers V. 16, John Alroy and Gene Hunt (eds.)

Value

A numeric value of estimated taxonomic richness

Author(s)

Andrew A. Zaffos

Examples

```
# Download a test dataset of Miocene-Pleistocene bivalves.
# DataPBDB<-downloadPBDB(Taxa="Bivalvia",StartInterval="Miocene",StopInterval="Pleistocene")

# Clean up the taxonomy by removing subgenus designation
# DataPBDB<-cleanTaxonomy(DataPBDB,"genus")

# Create a community matrix of genera by tectonic plate ids.
# CommunityMatrix<-abundanceMatrix(DataPBDB,Rows="geoplate",Columns="genus")

# Cull out depauperate samples and rare taxa
# CommunityCull<-cullMatrix(CommunityMatrix,5,100)

# Calculate the standardized richness of each plate at a quota of 0.5.
# StandardizedRichness<-apply(CommunityCull,1,subsampleEvenness,0.5)
```

subsampleIndividuals *Iterative Rarefaction*

Description

Calculate the richness of a sample after subsampling to a set number of individuals.

Usage

```
subsampleIndividuals(Abundance, Quota, Trials = 100)
```

Arguments

Abundance	A vector of taxon abundances
Quota	A whole number stating the desired sample size
Trials	Number of iterations

Details

This is an empirical approach to subsampling a vector of taxonomic abundances to a set number of abundances. It uses a bootstrapping approach rather than the more common analytical solution provided in other packages.

Value

A numeric value of estimated richness

Author(s)

Andrew A. Zaffos

Examples

```
# Download a test dataset of Miocene-Pleistocene bivalves
# DataPBDB<-downloadPBDB(Taxa="Bivalvia",StartInterval="Miocene",StopInterval="Pleistocene")

# Clean up the taxonomy by removing subgenus designation
# DataPBDB<-cleanTaxonomy(DataPBDB,"genus")

# Create a community matrix of genera by tectonic plate ids
# CommunityMatrix<-abundanceMatrix(DataPBDB,Rows="geoplate",Columns="genus")

# Cull out depauperate samples and rare taxa
# CommunityCull<-cullMatrix(CommunityMatrix,5,100)

# Calculate the standardized richness of each plate assuming a fixed sample size of 100 occurrences
# StandardizedRichness<-apply(CommunityCull,1,subsampleIndividuals,100)
```

taxonAlpha

Additive Diversity Partitioning functions

Description

Functions for calculating alpha, beta, and gamma richness of a community matrix under the Additive Diversity partitioning paradigm of R. Lande.

Usage

taxonAlpha(CommunityMatrix)

meanAlpha(CommunityMatrix)

taxonBeta(CommunityMatrix)

sampleBeta(CommunityMatrix)

totalBeta(CommunityMatrix)

totalGamma(CommunityMatrix)

Arguments

CommunityMatrix

a matrix

Details

Takes a community matrix (see `presenceMatrix` or `abundanceMatrix`) and returns the either the alpha, beta, or gamma richness of a community matrix. Some of these functions were presented in Holland, SM (2010) "Additive diversity partitioning in palaeobiology: revisiting Sepkoski's question" *Paleontology* 53:1237-1254. Namely, `taxonAlpha`, `taxonBeta`, `sampleBeta`, `meanAlpha`, `totalBeta`, and `totalGamma`.

- `taxonAlpha(CommunityMatrix)` Calculates the contribution to alpha diversity of each taxon.
- `meanAlpha(CommunityMatrix)` Calculates the average alpha diversity of all samples.
- `taxonBeta(CommunityMatrix)` Calculates the contribution to beta diversity of each taxon.
- `sampleBeta(CommunityMatrix)` Calculates the contribution to beta diversity of each sample.
- `totalBeta(CommunityMatrix)` Calculates the total beta diversity.
- `totalGamma(CommunityMatrix)` Calculates the richness of all samples in the community matrix.

Value

A vector of the alpha, beta, or gamma richness of a taxon, sample, or entire community matrix.

Author(s)

Andrew A. Zaffos

Examples

```
# Download a test dataset of pleistocene bivalves.
# DataPBDB<-downloadPBDB(Taxa="Bivalvia",StartInterval="Pleistocene",StopInterval="Pleistocene")

# Create a community matrix with tectonic plates as "samples"
# CommunityMatrix<-abundanceMatrix(DataPBDB,"geoplate")

# Calculate the average richness of all samples in a community.
# meanAlpha(CommunityMatrix)

# The beta diversity of all samples in a community.
# totalBeta(CommunityMatrix)

# This is, by definition, equivalent to the gamma diversity - mean alpha diversity.
# totalBeta(CommunityMatrix)==(totalGamma(CommunityMatrix)-meanAlpha(CommunityMatrix))
```

uniformExtinction	<i>Uniform Confidence Intervals</i>
-------------------	-------------------------------------

Description

Estimate a confidence interval on time of extinction or origination when assuming a uniform probability of collection.

Usage

```
uniformExtinction(Ages, Confidence = 0.95)
```

```
uniformOrigination(Ages, Confidence = 0.95)
```

Arguments

Ages a numeric vector of fossil occurrence ages, expressed as millions of years ago.

Confidence the desired confidence level

Details

Takes a numeric vector of fossil occurrence ages, expressed as millions of years ago, to estimate either the time of origination `uniformOrigination` or extinction `uniformExtinction`. This code uses Marshall's adaptation of the function by Strauss and Sadler.

Marshall, C.R. (1990) "Confidence intervals on stratigraphic ranges" *Paleobiology* 16:1-10.

Value

A matrix listing the earliest and latest estimate for extinction or origination.

Author(s)

Andrew A. Zaffos

Examples

```
# Generate an example dataset of fossil ages
FakeAges<-runif(50,50,100)

# Calculate the inferred age of extinction with 95% confidence.
uniformExtinction(FakeAges,Confidence=0.95)

# Calculate the inferred age of origination with 50% confidence.
uniformOrigination(FakeAges,Confidence=0.5)
```

Index

abundanceMatrix, 2
adaptiveExtinction
 (adaptiveOrigination), 3
adaptiveExtinction, adaptiveOrigination, thetaNegative, thetaPositive, lambdaNegative, lambdaPositive
 (adaptiveOrigination), 3
adaptiveOrigination, 3
ageRanges, 4

cleanTaxonomy, 5
completeTurnovers (multiplicativeBeta),
 11
constrainAges, 6
constrainAges, multiplyAges
 (constrainAges), 6
cullMatrix, 7
cullMatrix, errorMatrix, culltaxa, cullSamples, occurrencesFlag, diversityFlag, softCull, softTaxa, softSamples
 (cullMatrix), 7

downloadPaleogeography, 8
downloadPBDB, 9
downloadTime, 10

meanAlpha (taxonAlpha), 15
multiplicativeBeta, 11
multiplicativeBeta, completeTurnovers, notEndemic
 (multiplicativeBeta), 11
multiplyAges (constrainAges), 6

notEndemic (multiplicativeBeta), 11

presenceMatrix, 12

sampleBeta (taxonAlpha), 15
subsampleEvenness, 13
subsampleIndividuals, 14

taxonAlpha, 15
taxonAlpha, meanAlpha, taxonBeta, sampleBeta, totalGamma
 (taxonAlpha), 15
taxonBeta (taxonAlpha), 15
totalBeta (taxonAlpha), 15
totalGamma (taxonAlpha), 15
uniformExtinction, 17
uniformExtinction, uniformOrigination, lambdaNegative, lambdaPositive
 (uniformExtinction), 17
uniformOrigination (uniformExtinction),
 17