Package 'velociraptr'

October 12, 2022

Type Package

Title Fossil Analysis

Version 1.1.0

Author Andrew A. Zaffos

Maintainer Andrew A Zaffos <azaffos@email.arizona.edu>

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

Imports sf

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-07-31 19:20:02 UTC

R topics documented:

---

abundanceMatrix   *Create a community matrix of taxon abundances*

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 characer string

Columns   A character string

Details

Note that older versions of this function automatically checked for and removed hanging factors. However, this is something that should really be dictated by the user, and that step is no longer a part of the function. This is unlikely to introduce any breaking changes in older scripts, but we note it here for documentation purposes..

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")
```

---

ageRanges   *Find the age range for each taxon in a dataframe*

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 characer string identifying the desired level of the taxonomic hierarchy.

Details

Returns a data frame of that states gives the time of origination and extinction for each taxon as numeric values. Note that older versions of this function automatically dropped hanging factors and NA's, but that cleaning step should ideally be dictated by the user up-front. So that functionality has been dropped. This may introduce breaking chanes in legacy scripts, but is easily fixed by standard data cleaning steps.

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   *Clean taxonomic names*

Description

Removes NAs and subgenera from the genus column.

Usage

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

Arguments

Data   A data frame of taxonomic ocurrences 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   *Constrain a dataset to only occurrences within a certain age-range*

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

cullMatrix	*Cull rare taxa and depauperate samples*

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 sampe 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

A simple features object

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.

- ocurrence_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 occurence 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")
```

---

downloadPlaces   *Download Shapefile of Places*

Description

Download a shapefile of a geolocation from the Macrostrat API's implementation of the Who's on First database by MapZen.

Usage

```
downloadPlaces(Place = "Wisconsin", Type = "region")
```

Arguments

Place	A character string; the name of a place
Type	A character string; a type of place

Details

Download a shapefile of a geolocation from the Macrostrat API. The Macrostrat database provides a GeoJSON of a particular location given the location's name and type. Type can be of the categories: "continent", "country", "region", "county", and "locality".

If multiple locations of the same type share the same name (e.g., Alexandria), the route will return a feature collection of all matching polygons.

Value

An rgdal compatible shapefile

Author(s)

Andrew A. Zaffos

Examples

```
# Download a polygon of Dane County, Wisconsin, United States, North America
# DaneCounty<-downloadPlaces(Place="Dane",Type="county")

# Download a polygon of Wisconsin, United States, North America
# Wisconsin<-downloadPlaces(Place="Wisconsin",Type="region")

# Download a polygon of North America
# NorthAmerica<-downloadPlaces(Place="North America",Type="continent")
```

---

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")
```

---

fixedLatitude	*Download fixed-latitude equal-area grid*

Description

Download an equal-area grid of the world with fixed latitudinal spacing and variable longitudinal spacing.

Usage

```
fixedLatitude(LatSpacing = 5, CellArea = "500000")
```

Arguments

LatSpacing	Number of degrees desired between latitudinal bands
CellArea	Desired target area of the cells in km^2 as a character string

Details

Downloads an equal-area grid with fixed latitudinal spacing and variable longitudinal spacing. The distance between longitudinal borders of grids will adjust to the target area size within each band of latitude. The algorithm will adjust the area of the grids to ensure that the total surface of the globe is covered.

Value

A simple features object

Author(s)

Andrew A. Zaffos

Examples

```
# Download an equal area grid with 10 degree latitudinal spacing and 1,000,000 km^2 grids
# EqualArea<-fixedLatitude(LatSpacing=10,CellArea="1000000")
```

---

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.

Refer to 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 - Gamma/Alpha. It quantifies how many times as rich gamma is than alpha.

- `completeTurnovers(CommunityMatrix)`: The number of complete effective species turnovers observed among compositonal units in the dataset - (Gamma-Alpha)/Alpha.

- `notEndemic(CommunityMatrix)`: The proportion of taxa in the dataset not limited to a single sample - (Gamma-Alpha)/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 characer 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")
```

---

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.

These functions were originally presented in Holland, SM (2010) "Additive diversity partitioning in palaeobiology: revisiting Sepkoski's question" *Paleontology* 53:1237-1254.

- 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))
```

Index