Package ‘paleobioDB’

July 25, 2020

Description Includes 19 functions to wrap each endpoint of
the PaleobioDB API, plus 8 functions to visualize and process the fossil
data. The API documentation for the Paleobiology Database can be found in
<http://paleobiodb.org/data1.1/>.

Encoding UTF-8

Type Package

License GPL-2

Title Download and Process Data from the Paleobiology Database

Version 0.7.0

Date 2020-07-22

Maintainer Sara Varela <svarela@paleobiogeography.org>

BugReports https://github.com/ropensci/paleobioDB/issues

URL https://docs.ropensci.org/paleobioDB,
https://github.com/ropensci/paleobioDB

LazyData true

Imports rjson, gtools, RCurl, plyr, scales

Depends raster, maps

Suggests roxygen2, testthat

Collate 'network.R' 'paleobioDB-package.R' 'pbdb_cache.R'
'pbdb_geographic_functions.R' 'pbdb_querys.R'
'rest_api_tools.R' 'pbdb_rest_api_setup.R'
'pbdb_taxonomic_functions.R' 'pbdb_temporal_functions.R'

RoxygenNote 5.0.1

NeedsCompilation no

Author Sara Varela [aut, cre],
Javier González Hernández [aut],
Luciano Fabris Sgarbi [aut]

Repository CRAN

Date/Publication 2020-07-25 21:40:02 UTC

R topics documented:

---

paleobioDB *paleobioDB: An R-package for downloading, visualizing and processing data from the Paleobiology Database*

Description

We have developed paleobioDB, an R-package designed to make easy and flexible queries of the Paleobiology Database, as well as to visualize and download selected data. This package will make it easy to access paleontological data in a way that should allow those data to be further analyzed, including via packages and libraries available in R.

Details

We programmed two different groups of functions. First, we developed a set of general and flexible functions to wrap the PaleobioDB API. These functions connect R with each of the endpoints of the PaleobioDB API. Second, based on these base functions, we programmed a second set of functions intended to explore and visualize the fossil occurrences in their geographic, temporal and taxonomic dimensions.

Package:	paleobioDB
Type:	Package
Version:	0.4
Date:	2015-07-16
License:	GPL-2

Author(s)

Sara Varela <svarela@paleobiogeography.org>

Javier Gonzalez <javigzz@yahoo.es>

Luciano Fabris Sgarbi <luciano.f.sgarbi@gmail.com>

References

Sara Varela, Javier Gonzalez-Hernandez, Luciano Fabris Sgarbi, Charles Marshall, Mark D. Uhen, Shanan Peters, Michael McClennen, 2014. paleobioDB: an R-package for downloading, visualizing and processing data from the Paleobiology Database (under review)

See Also

http://paleobiodb.org

Examples

```
## Not run:

canidae<-  pbdb_occurrences (limit="all", base_name="canidae",
interval="Quaternary", show=c("coords", "phylo", "ident"))

## to explore the number of subtaxa
pbdb_subtaxa (canidae)

## to explore the temporal resolution of the fossil records
pbdb_temporal_resolution (canidae)

## returns a dataframe and a plot with the temporal span
##  of the species, genera, etc.
pbdb_temp_range (canidae, rank= "genus", names=FALSE)

## returns a dataframe and a plot showing the species, genera, etc.
richness across time
pbdb_richness (canidae, rank= "species",
temporal_extent= c (0,10), res= 1)

## returns a dataframe and a plot showing the evolutionary
and extinction rates across time

## evolutionary rates= evo_ext=1
pbdb_orig_ext (canidae, rank="species", temporal_extent=c(0, 10),
res=1, orig_ext=1)

## extinction rates= evo_ext=2
pbdb_orig_ext (canidae, rank="species", temporal_extent=c(0, 10),
              res=1, orig_ext=2)

## maps the fossil occurrences
pbdb_map (canidae, main = "Canidae", pch= 19, cex=0.7)

## maps the sampling effort
pbdb_map_occur (canidae, res= 5)

## maps the species, genera, etc. richness
pbdb_map_richness (canidae, rank="species", res= 5)


## End(Not run)
```

pbdb_collection *pbdb_collection*

Description

Returns information about a single collection record from the Paleobiology Database.

Usage

```
pbdb_collection (id, ...)
```

Arguments

`id`	identifier of the collection. This parameter is required.
`...`	additional arguments passed to the API. See all available arguments in `http://paleobiodb.org/data1.1/colls/single`. Eg: • *vocab*: set vocab="pbdb" to show the complete name of the variables (by default variables have short 3-letter names) • *show*: show extra variables • ...

Details

Go to pbdb_occurrences to see an explanation about the main filtering parameters.

Value

a dataframe with a single occurrence

Examples

```
## Not run:
pbdb_collection (id=1003, vocab="pbdb", show="loc")


## End(Not run)
```

---

pbdb_collections *pbdb_collections*

Description

Returns information about multiple collections, selected according to the parameters you provide.

Usage

```
pbdb_collections (...)
```

Arguments

...	documentation for all the parameters is available in http://paleobiodb.org/data1.1/colls/list go to pbdb_occurrences to see an explanation about the main filtering parameters

Value

a dataframe with the collections that match the query

Examples

```
## Not run:
pbdb_collections (base_name="Cetacea", interval="Miocene")

## End(Not run)
```

pbdb_collections_geo *pbdb_collections_geo*

Description

This path returns information about geographic clusters of collections from the Paleobiology Database. These clusters are defined in order to facilitate the generation of maps at low resolutions. You can make a config request via http://paleobiodb.org/data1.1/config in order to get a list of the available summary levels.

Usage

```
pbdb_collections_geo (...)
```

Arguments

`...`	documentation for all the parameters is available in http://paleobiodb.org/data1.1/colls/summary go to `pbdb_occurrences` to see an explanation about the main filtering parameters

Value

a dataframe with the collections that match the query

Examples

```
## Not run:
pbdb_collections_geo (vocab="pbdb", lngmin=0.0,
lngmax=15.0, latmin=0.0, latmax=15.0, level=2)

## End(Not run)
```

pbdb_interval *pbdb_interval*

Description

Returns information about a single interval, selected by identifier.

Usage

```
pbdb_interval (id, ...)
```

Arguments

`id` identifier of the temporal interval. This parameter is required.

`...` additional arguments passed to the API. See documentation for accepted parameters in http://paleobiodb.org/data1.1/intervals/single. Eg:

- *vocab*: set vocab="pbdb" to show the complete name of the variables (by default variables have short 3-letter names)

Value

a dataframe with information from a single temporal interval

Examples

```
## Not run:
pbdb_interval (id=1, vocab="pbdb")

## End(Not run)
```

pbdb_intervals *pbdb_intervals*

Description

Returns information about multiple intervals, selected according to the parameters you provide.

Usage

```
pbdb_intervals (...)
```

Arguments

`...` arguments passed to the API. See documentation for accepted parameters in http://paleobiodb.org/data1.1/intervals/list. Eg:

- *min_ma*: return only intervals that are at least this old
- *max_ma*: return only intervals that are at most this old
- *order*: return the intervals in order starting as specified. Possible values include older, younger. Defaults to younger
- *vocab*: set vocab="pbdb" to show the complete name of the variables (by default variables have short 3-letter names)
- ...

Value

a dataframe with information from several temporal intervals

Examples

```
## Not run:
pbdb_intervals (min_ma= 0, max_ma=2, vocab="pbdb")

## End(Not run)
```

---

pbdb_map *pbdb_map*

Description

Maps the fossil records

Usage

```
pbdb_map (data, col.int="white" ,pch=19, col.ocean="black", main=NULL,
col.point=c("light blue","blue"), ...)
```

Arguments

`data`	Input dataframe. This dataframe is the output of `pbdb_occurrences` function using the argument: `show = "coords"`. See too: **Details** and **Examples**
`col.int`	The colour of the mainland.
`pch`	See: `par`
`col.ocean`	The colour of the ocean.
`main`	To set the title of the map. See: `par`
`col.point`	Two or more colours. To generate the colour gradient used to show the number of occurrences per cell in map
`...`	Others parameters. See `par` and `map`

Details

The function opens a new window for the map

CAUTION! The argument `show = "coords"` in `pbdb_occurrences` function is required. We recommend the use of a cairo device (`X11`) for better visualization of the graphs. See **Examples**

Value

A map showing the distribution of the fossil records, with the points with a color gradient, according to the number of occurrences per cell.

See Also

See `pbdb_occurrences`, `map`, `par` and `colors` help pages

Examples

```
## Not run:
data<- pbdb_occurrences (limit="all", vocab= "pbdb",
base_name="Canis", show="coords")
X11(width=12, height=8)
pbdb_map(data)
pbdb_map(data,pch=1)
pbdb_map(data,pch=19,col.point=c("pink","red"), col.ocean="light blue",
main="canis")

## End(Not run)
```

---

pbdb_map_occur *pbdb_map_occur*

Description

Creates a RasterLayer object and a plot of the sampling effort (number of fossil records per cell).

Usage

```
pbdb_map_occur (data, res=5, col.int="white", col.ocean="black",
col.eff=c("light blue","blue"), do.plot=TRUE, ...)
```

Arguments

`data`	Input dataframe. This dataframe is the output of `pbdb_occurrences` function using the argument: `show="coords"`. See too: **Details** and **Examples**
`res`	the resolution of the RasterLayer object (in decimal degrees). See: `raster`
`col.int`	The colour of the mainland
`col.ocean`	The colour of the ocean
`col.eff`	Two or more colours. To generate the colour gradient used to show the number of occurrences per cell in map
`do.plot`	Logical; `TRUE` the function returns a RasterLayer and a plot.
`...`	Others parameters. See `par` and `map`

Details

CAUTION! The argument `show = "coords"` in `pbdb_occurrences` function is required. We recommend the use of a cairo device (`X11`) for better visualization of the graphs. See **Examples**

Value

A RasterLayer object and a plot with the sampling effort (number of fossil records per cell). This RasterLayer object have the resolution controlled by the argument `res`. The deflaut is `res=1`.

See Also

See pbdb_occurrences, map, par and colors help pages

Examples

```
## Not run:
data<- pbdb_occurrences (limit="all", vocab= "pbdb", base_name="Canis",
show="coords")
X11(width=13, height=7.8)
pbdb_map_occur (data,res=2)
## to obtain the raster file without plotting it
pbdb_map_occur (data,res=3,do.plot=F)

## End(Not run)
```

pbdb_map_richness *pbdb_map_richness*

Description

Creates a RasterLayer object and a plot with richness of species, genera, families, etc. per cell.

Usage

```
pbdb_map_richness (data, rank="species", do.plot=TRUE, res=5,
col.int="white", col.ocean="black",
col.rich=c("light blue","blue"),...)
```

Arguments

data	Input dataframe. This dataframe is the output of pbdb_occurrences function using the argument: show = c("phylo","coords","ident"). See too: **Details** and **Examples**
rank	To set which taxon rank you are interested for calculate richness. The options are: "species", "genus", "family", "order", "class" or "phylum")
do.plot	Logical; TRUE the function returns a RasterLayer and a plot.
res	The resolution of the RasterLayer object (in decimal degrees). See: raster
col.int	The colour of the mainland
col.ocean	The colour of the ocean
col.rich	Two or more colours. To generate the colour gradient used to show the richness per cell in map
...	Others parameters. See par and map

Details

CAUTION! The argument show = "coords" in pbdb_occurrences function is required. We recommend the use of a cairo device (X11) for better visualization of the graphs. See **Examples**

Value

A RasterLayer object and a plot with richness of species, genera, families, etc. per cell. This RasterLayer object have the resolution controlled by the argument res. The default is res=1.

See Also

See pbdb_occurrences, map, par and colors help pages

Examples

```
## Not run:
data<- pbdb_occurrences (limit=1000, vocab= "pbdb", base_name="mammalia",
show=c("phylo","coords","ident"))
X11(width=13, height=7.8)
pbdb_map_richness (data,res=8,rank="genus")
pbdb_map_richness (data,res=8,rank="family")
## to obtain the raster file and not plot the map
pbdb_map_richness (data,res=8,rank="family",do.plot=F)

## End(Not run)
```

---

pbdb_occurrence *pbdb_occurrence*

Description

Returns information about a single occurrence record from the Paleobiology Database.

Usage

```
pbdb_occurrence (id, ...)
```

Arguments

id	identifier of the occurrence. This parameter is required
...	arguments passed to the API. See all available arguments in http://paleobiodb.org/data1.1/occs/single. Eg: • *vocab*: set vocab="pbdb" to show the complete name of the variables (by default variables have short 3-letter names)

Details

Documentation for all the parameters is available at http://paleobiodb.org/data1.1/occs/single. In the parameter list above, we describe the most common filters that paleontologists and ecologists might use.

Value

a dataframe with a single occurrence

Examples

```
## Not run:
pbdb_occurrence (id=1001)
pbdb_occurrence (id=1001, vocab="pbdb", show="coords")

## End(Not run)
```

---

pbdb_occurrences *pbdb_occurrences*

Description

Returns information about species occurrence records stored in the Paleobiology Database.

Usage

```
pbdb_occurrences(...)
```

Arguments

`...` arguments passed to the API. See all available arguments in http://paleobiodb.org/data1.1/occs/list

- *limit*: sets limit to "all" to download all the occurrences. By default the limit is 500.
- *taxon_name*: Return only records associated with the specified taxonomic name(s). You may specify multiple names, separated by commas.
- *base_name*: Return records associated with the specified taxonomic name(s) and any of their children (e.g. base_name="Canis" will
- *lngmin*: numeric. The longitude boundaries will be normalized to fall between -180 and 180. Note that if you specify lngmin then you must also specify lngmax. Returns only records whose geographic location falls within the given bounding box (defined by lngmin, lngmax, latmin, latmax). It generates two adjacent bounding boxes if the range crosses the antimeridian.
- *lngmax*: numeric. The longitude boundaries will be normalized to fall between -180 and 180.

- *latmin*: numeric. between -90 and 90. Note that if you specify latmin then you must also specify latmax.
- *latmax*: numeric. between -90 and 90.
- *min_ma*: return only records whose temporal locality is at least this old, specified in Ma.
- *max_ma*: return only records whose temporal locality is at most this old, specified in Ma.
- *interval*: return only records whose temporal locality falls within the named geologic time interval (e.g. "Miocene").
- *continent*: return only records whose geographic location falls within the specified continent(s).
- *show*: to show extra variables (e.g. coords, phylo, ident)

Details

Documentation for all the parameters is available at http://paleobiodb.org/data1.1/occs/list. We describe the most common filters that paleontologists and ecologists might use in the parameter list above.

Value

a dataframe with the species occurrences

Examples

```
## Not run:
pbdb_occurrences (id=c(10, 11), show=c("coords", "phylo", "ident"))
pbdb_occurrences (limit="all", vocab= "pbdb",
taxon_name="Canis", show=c("coords", "phylo", "ident"))
pbdb_occurrences (limit="all", vocab= "pbdb",
base_name="Canidae", show=c("coords", "phylo", "ident"))

## End(Not run)
```

pbdb_orig_ext *pbdb_orig_ext*

Description

Plots the appearance of new taxa across time.

Usage

```
pbdb_orig_ext (data, rank,
temporal_extent, res, orig_ext,
colour="#0000FF30", bord="#0000FF", do.plot=TRUE)
```

Arguments

`data` dataframe with our query to the paleoBD `pbdb_occurrences`. Important, it is required to show the name of the families, orders, etc. in the dataframe, to do that set: show=c("phylo", "ident") (see example).

`rank` to set which taxon rank you are interested. By default rank= "species"

`temporal_extent` vector to set the temporal extent (min, max)

`res` numeric. to set the intervals of the temporal extent

`orig_ext` 1= origination, 2=extinction.

`colour` to change the colour of the bars in the plot, skyblue2 by default.

`bord` to set the colour of the border of the polygon

`do.plot` TRUE/FALSE (TRUE by default).

Value

a dataframe with the number of first appearances and extinctions of the selected taxon rank across time, and a plot with the first appearances or extinctions of the selected taxon rank across time.

Examples

```
## Not run:
canidae<-  pbdb_occurrences (limit="all", vocab="pbdb",
base_name="Canidae", show=c("phylo", "ident"))

# plot of the evolutive rates.
pbdb_orig_ext (canidae, rank="genus", temporal_extent=c(0, 10),
res=1, orig_ext=1)

# plot of the extinction rates.
pbdb_orig_ext (canidae, rank="species", temporal_extent=c(0, 10),
res=1, orig_ext=2)

## End(Not run)
```

`pbdb_reference` *pbdb_reference*

Description

Returns information about a single reference, selected by identifier. Go to `pbdb_occurrences` to see an explanation about the main filtering parameters

Usage

```
pbdb_reference (id, ...)
```

Arguments

`id` identifier of the reference. This parameter is required.

`...` arguments passed to the API. See documentation for accepted parameters in http://paleobiodb.org/data1.1/refs/single. Eg:

- *vocab*: set vocab="pbdb" to show the complete name of the variables (by default variables have short 3-letter names)
- ...

Value

a dataframe with a single reference

Examples

```
## Not run:
pbdb_collection (id=1003, vocab="pbdb", show="loc")

## End(Not run)
```

---

pbdb_references *pbdb_references*

Description

Returns information about multiple references, selected according to the parameters you provide.

Usage

```
pbdb_references (...)
```

Arguments

`...` arguments passed to the API. See documentation for accepted parameters in http://paleobiodb.org/data1.1/refs/list. Eg:

- *author* select only references for which any of the authors matches the specified name
- *year* select only references published in the specified year
- *pubtitle* select only references that involve the specified publication
- *order* specifies the order in which the results are returned. You can specify multiple values separated by commas, and each value may be appended with .asc or .desc. Accepted values are: author, year, pubtitle, created, modified, rank.
- ...

Value

a dataframe with the information about the references that match the query

Examples

```
## Not run:
  pbdb_references (author= "Polly")

## End(Not run)
```

---

pbdb_ref_collections *pbdb_ref_collections*

Description

Returns information about the references from which the selected collection data were entered.

Usage

```
pbdb_ref_collections (...)
```

Arguments

`...`	arguments passed to the API. See documentation for accepted parameters in `http://paleobiodb.org/data1.1/colls/refs`. Eg: - *id* comma-separated list of collection identifiers - *author* select only references for which any of the authors matches the specified name - *year* select only references published in the specified year - *pubtitle* select only references that involve the specified publication - *order* specifies the order in which the results are returned. You can specify multiple values separated by commas, and each value may be appended with .asc or .desc. Accepted values are: author, year, pubtitle, created, modified, rank. - ...

Value

a dataframe with the information about the references that match the query

Examples

```
## Not run:
  pbdb_ref_collections (id=1)

## End(Not run)
```

pbdb_ref_occurrences *pbdb_ref_occurrences*

Description

Returns information about the bibliographic references associated with fossil occurrences from the database.

Usage

```
pbdb_ref_occurrences (...)
```

Arguments

`...`	arguments passed to the API. See all available arguments in `http://paleobiodb.org/data1.1/occs/refs` - *author* select only references for which any of the authors matches the specified name - *year* select only references published in the specified year - *pubtitle* select only references that involve the specified publication - *order* specifies the order in which the results are returned. You can specify multiple values separated by commas, and each value may be appended with .asc or .desc. Accepted values are: author, year, pubtitle, created, modified, rank.

Details

Go to `pbdb_occurrences` to see an explanation about the main filtering parameters.

Value

a dataframe with the information about the references that match the query

Examples

```
## Not run:
pbdb_ref_occurrences (vocab="pbdb",
base_name="Canis", year=2000)

## End(Not run)
```

`pbdb_ref_taxa` *pbdb_ref_taxa*

Description

This URL path returns information about the source references associated with taxa in the Paleobiology Database. You can use the same parameters that are available with pbdb_taxa, but Reference records are returned instead of Taxon records. One record is returned per reference, even if it is associated with multiple taxa.

Usage

```
pbdb_ref_taxa (...)
```

Arguments

`...` arguments passed to the API. See all available arguments in http://paleobiodb.org/data1.1/taxa/refs

- *name*: returns information about the most fundamental taxonomic name matching this string. The % and _ characters may be used as wildcards.
- *id*: returns information about the taxonomic name corresponding to this identifier. You may not specify both name and id in the same query.
- *exact*: if this parameter is specified, then the taxon exactly matching the specified name or identifier is selected, rather than the senior synonym which is the default.
- *show*: show extra variables
- *rel*: set rel="synonyms" to select all synonyms of the base taxon or taxa; rel="children" to select the taxa immediately contained within the base taxon or taxa; rel="common_ancestor" to select the most specific taxon that contains all of the base taxa.
- *extant*: TRUE/FALSE to select extant/extinct taxa.

Value

a dataframe with references from a list of taxa

Examples

```
## Not run:
  pbdb_ref_taxa (name="Canidae", vocab="pbdb", show=c("attr", "app", "size", "nav"))

## End(Not run)
```

pbdb_richness *pbdb_richness*

Description

Plots the number of the interested.

Usage

```
pbdb_richness (data, rank, res, temporal_extent, colour, bord, do.plot)
```

Arguments

data	dataframe with our query to the paleoBD pbdb_occurrences. Important, it is required to show the name of the families, orders, etc. in the dataframe, to do that set: show=c("phylo", "ident") (see example).
rank	to set which taxon rank you are interested. By default rank= "species"
res	numeric. to set the intervals of the temporal extent
temporal_extent	vector to set the temporal extent (min, max)
colour	to change the colour of the bars in the plot, skyblue2 by default.
bord	to set the colour of the border of the polygon
do.plot	TRUE/FALSE (TRUE by default).

Value

a plot and a dataframe with the richness aggregated by the taxon rank in the specified temporal extent and resolution.

Examples

```
## Not run:
data<-  pbdb_occurrences (limit="all", vocab="pbdb",
base_name="Canidae", show=c("phylo", "ident"))
pbdb_richness (data, rank="species", res=1, temporal_extent=c(0,3))

## End(Not run)
```

pbdb_scale *pbdb_scale*

Description

Returns information about a single time scale, selected by identifier.

Usage

```
pbdb_scale (id, ...)
```

Arguments

id	identifier of the temporal interval. This parameter is required.
...	additional arguments passed to the API. See documentation for accepted parameters in http://paleobiodb.org/data1.1/scales/single. Eg: • *vocab*: set vocab="pbdb" to show the complete name of the variables (by default variables have short 3-letter names) • ...

Value

a dataframe with information from a single scale

Examples

```
## Not run:
  pbdb_scale (id=1, vocab="pbdb")

## End(Not run)
```

pbdb_scales *pbdb_scales*

Description

Returns information about multiple time scales.

Usage

```
pbdb_scales(...)
```

Arguments

`...` arguments passed to the API. See documentation for accepted parameters in http://paleobiodb.org/data1.1/scales/list. Eg:

- *vocab*: set vocab="pbdb" to show the complete name of the variables (by default variables have short 3-letter names)
- ...

Value

a dataframe with information from the selected scales

Examples

```
## Not run:
## Get a dataframe with all the scales available in PBDB
## by setting no ids
pbdb_scales ()

## End(Not run)
```

pbdb_strata *pbdb_strata*

Description

Returns information about geological strata, selected by name, rank, and/or geographic location.

Usage

```
pbdb_strata (...)
```

Arguments

`...` arguments passed to the API. See documentation for accepted parameters in http://paleobiodb.org/data1.1/strata/list. Eg:

- *name*: a full or partial name. You can use % and _ as wildcards, but the query will be very slow if you put a wildcard at the beginning
- *rank*: returns only strata of the specified rank: formation, group or member.
- *lngmin*: numeric. The longitude boundaries will be normalized to fall between -180 and 180. Note that if you specify lngmin then you must also specify lngmax. Returns only records whose geographic location falls within the given bounding box (defined by lngmin, lngmax, latmin, latmax). It generate two adjacent bounding boxes if the range crosses the antimeridian.
- *lngmax*: numeric. The longitude boundaries will be normalized to fall between -180 and 180.

- *latmin*: numeric. between -90 and 90. Note that if you specify latmin then you must also specify latmax.
- *latmax*: numeric. between -90 and 90.
- *loc*: Return only strata associated with some occurrence whose geographic location falls within the specified geometry, specified in WKT format.
- *vocab*: set vocab="pbdb" to show the complete name of the variables (by default variables have short 3-letter names)
- ...

Value

a dataframe with information from the selected strata

Examples

```
## Not run:
pbdb_strata (lngmin=0, lngmax=15, latmin=0, latmax=15, rank="formation", vocab="pbdb")

## End(Not run)
```

---

pbdb_strata_auto *pbdb_strata_auto*

Description

Returns a list of strata matching the given prefix or partial name. This can be used to implement auto-completion for strata names, and can be limited by geographic location if desired.

Usage

```
pbdb_strata_auto (...)
```

Arguments

`...` arguments passed to the API. See documentation for accepted parameters in `http://paleobiodb.org/data1.1/strata/auto`. Eg:

- *name*: a full or partial name. You can use % and _ as wildcards, but the query will be very slow if you put a wildcard at the beginning
- *rank*: returns only strata of the specified rank: formation, group or member.
- *lngmin*: numeric. The longitude boundaries will be normalized to fall between -180 and 180. Note that if you specify lngmin then you must also specify lngmax. Returns only records whose geographic location falls within the given bounding box (defined by lngmin, lngmax, latmin, latmax). It generate two adjacent bounding boxes if the range crosses the antimeridian.
- *lngmax*: numeric. The longitude boundaries will be normalized to fall between -180 and 180.

- *latmin*: numeric. between -90 and 90. Note that if you specify latmin then you must also specify latmax.
- *latmax*: numeric. between -90 and 90.
- *loc*: Return only strata associated with some occurrence whose geographic location falls within the specified geometry, specified in WKT format.
- *vocab*: set vocab="pbdb" to show the complete name of the variables (by default variables have short 3-letter names)
- ...

Value

a dataframe with information from the strata that matches our letters.

Examples

```
## Not run:
pbdb_strata_auto (name= "Pin", vocab="pbdb")

## End(Not run)
```

---

pbdb_subtaxa *pbdb_subtaxa*

Description

count the number of subtaxa within a given taxa. e.g. number of species within a genus.

Usage

```
pbdb_subtaxa (data, do.plot, col)
```

Arguments

data	dataframe with our query to the paleoBD `pbdb_occurrences`
do.plot	by default this function make a plot to visualize the distribution of taxa. Set to FALSE to skip the plot.
col	set the colour of the histogram. skyblue2 by default.

Value

a plot and a dataframe with the number of subtaxa in the data.

Examples

```
## Not run:
canidae_quat<-  pbdb_occurrences (limit="all",
base_name="Canidae",  interval="Quaternary",
show=c("coords", "phylo", "ident"))
pbdb_subtaxa (canidae_quat)

## End(Not run)
```

---

pbdb_taxa *pbdb_taxa*

Description

Returns information about multiple taxonomic names. This function can be used to query for all of the children or parents of a given taxon, among other operations.

Usage

```
pbdb_taxa (...)
```

Arguments

`...` arguments passed to the API. See all available arguments in http://paleobiodb.org/data1.1/taxa/list

- *name*: returns information about the most fundamental taxonomic name matching this string. The % and _ characters may be used as wildcards.
- *id*: return information about the taxonomic name corresponding to this identifier. You may not specify both name and id in the same query.
- *exact*: if this parameter is specified, then the taxon exactly matching the specified name or identifier is selected, rather than the senior synonym which is the default.
- *show*: to show extra variables: *attr* the attribution of this taxon (author and year); *app* the age of first and last appearance of this taxon from the occurrences recorded in this database; *size* the number of subtaxa appearing in this database; *nav* additional information for the PBDB Navigator taxon browser
- *rel*: set rel="synonyms" to select all synonyms of the base taxon or taxa; rel="children" to select the taxa immediately contained within the base taxon or taxa; rel="common_ancestor" to select the most specific taxon that contains all of the base taxa.
- *extant*: TRUE/FALSE to select extant/extinct taxa.

Value

a dataframe with information from a list of taxa

Examples

```
## Not run:
pbdb_taxa (name="Canidae", vocab="pbdb",
show=c("attr", "app", "size", "nav"))
pbdb_taxa (id =c(10, 11), vocab="pbdb",
show=c("attr", "app", "size", "nav"))
pbdb_taxa (id =c(10, 11), vocab="pbdb",
show=c("attr", "app", "size", "nav"), rel="common_ancestor")

## End(Not run)
```

`pbdb_taxa_auto` *pbdb_taxa_auto*

Description

Returns a list of names matching the given prefix or partial name.

Usage

```
pbdb_taxa_auto (...)
```

Arguments

`...` arguments passed to the API. See documentation for accepted parameters in `http://paleobiodb.org/data1.1/taxa/auto_doc.html`. Eg:

- *name*: a partial name or prefix. It must have at least 3 significant characters, and may include both a genus (possibly abbreviated) and a species.
- *limit*: set the limit to the number of matches
- ...

Value

a dataframe with information about the matches (taxon rank and number of occurrences in the database)

Examples

```
## Not run:
pbdb_taxa_auto (name="Cani", limit=10)

## End(Not run)
```

`pbdb_taxon` *pbdb_taxon*

Description

Returns information about a single taxonomic name, identified either by name or by identifier.

Usage

```
pbdb_taxon (...)
```

Arguments

`...` arguments passed to the API. See documentation for accepted parameters in `http://paleobiodb.org/data1.1/taxa/single`. Eg:

- *name*: returns information about the most fundamental taxonomic name matching this string. The % and _ characters may be used as wildcards.
- ...

Value

a dataframe with information from a single taxon

Examples

```
## Not run:
pbdb_taxon (name="Canis", vocab="pbdb",
show=c("attr", "app", "size"))


## End(Not run)
```

`pbdb_temporal_resolution` *pbdb_temporal_resolution*

Description

to show the temporal resolution of the fossil data

Usage

```
pbdb_temporal_resolution (data, do.plot=TRUE)
```

Arguments

`data`	dataframe with our query to the paleoBD `pbdb_occurrences`
`do.plot`	TRUE/FALSE. To show a frequency plot of the data (TRUE by default).

Value

a plot and a list with a summary of the temporal resolution of the data (min, max, 1st and 3rd quartils, median and mean), and the temporal resolution of each fossil record (Ma).

Examples

```
## Not run:
data<- pbdb_occurrences (taxon_name= "Canidae", interval= "Quaternary")
pbdb_temporal_resolution (data)

## End(Not run)
```

`pbdb_temp_range` *pbdb_temp_range*

Description

constructs a plot and a dataframe with the temporal range of the taxa (species, genera, families, etc.) within in a selected higher taxon.

Usage

```
pbdb_temp_range (data, rank, col = "#0000FF",
names = TRUE, do.plot =TRUE)
```

Arguments

`data`	dataframe with our query to the paleoBD `pbdb_occurrences`. Important, it is required to show the name of the families, orders, etc. in the dataframe, to do that set: show=c("phylo", "ident") (see example).
`rank`	to set which taxon rank you are interested.
`col`	to change the colour of the bars in the plot, skyblue2 by default.
`names`	TRUE/FALSE (TRUE by default). To include or not the name of the taxa in the plot
`do.plot`	TRUE/FALSE (TRUE by default).

Value

a plot and a dataframe with the time span of the taxa selected (species, genus, etc.)

Examples

```
## Not run:
canis_quaternary<- pbdb_occurrences (limit="all", base_name="Canis",
                 interval="Quaternary", show=c("coords", "phylo", "ident"))
pbdb_temp_range (canis_quaternary, rank="species", names=FALSE)

## End(Not run)
```

Index