Package ‘occCite’

June 16, 2025

Type Package

Title Querying and Managing Large Biodiversity Occurrence Datasets

Version 0.6.0

Maintainer Hannah L. Owens <hannah.owens@gmail.com>

Description Facilitates the gathering of biodiversity occurrence data from disparate sources. Metadata is managed throughout the process to facilitate reporting and enhanced ability to repeat analyses.

License GPL-3

URL https://docs.ropensci.org/occCite/

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

Encoding UTF-8

LazyData true

Language en-US

Depends R (>= 3.5.0)

Suggests ape, bit64, covr, knitr, httr, rmarkdown, remotes, testthat, taxize (>= 0.10)

Imports bib2df, BIEN, curl, dplyr, lubridate, methods, rgbif (>= 3.1), RefManageR, stringr, stats, leaflet, htmltools, ggplot2, rlang, tidyr, RPostgreSQL, RColorBrewer, viridis, DBI, waffle

VignetteBuilder knitr, rmarkdown

RoxygenNote 7.3.2

NeedsCompilation no

Author Hannah L. Owens [aut, cre] (ORCID: <https://orcid.org/0000-0003-0071-1745>),
Cory Merow [aut] (ORCID: <https://orcid.org/0000-0003-0561-053X>),
Brian Maitner [aut] (ORCID: <https://orcid.org/0000-0002-2118-9880>),
Jamie M. Kass [aut] (ORCID: <https://orcid.org/0000-0002-9432-895X>),
Vijay Barve [aut] (ORCID: <https://orcid.org/0000-0002-4852-2567>),
Robert P. Guralnick [aut] (ORCID: <https://orcid.org/0000-0001-6682-1504>),

Damiano Oldoni [rev] (ORCID: <https://orcid.org/0000-0003-3445-7562>, Damiano reviewed the package (v. 0.5.2) for rOpenSci, see <https://github.com/ropensci/software-review/issues/407>)

Repository CRAN

Date/Publication 2025-06-16 11:20:02 UTC

Contents

`GBIFLogin-class` *GBIFLogin Data Class*

Description

A class for managing GBIF login data.

Slots

`username` A vector of type character specifying a GBIF username.

`email` A vector of type character specifying the email associated with a GBIF username.

`pwd` A vector of type character containing the user's password for logging in to GBIF.

Examples

```
GBIFLogin <- GBIFLoginManager(
  user = "occCiteTester",
  email = "****@yahoo.com",
  pwd = "12345"
)
```

`GBIFLoginManager` *GBIF Login Manager*

Description

Takes users GBIF login particulars and turns it into a `GBIFLogin` for use in downloading data from GBIF. You MUST ALREADY HAVE AN ACCOUNT at GBIF.

Usage

```
GBIFLoginManager(user = NULL, email = NULL, pwd = NULL)
```

Arguments

`user`	A vector of type character specifying a GBIF username.
`email`	A vector of type character specifying the email associated with a GBIF username.
`pwd`	A vector of type character containing the user's password for logging in to GBIF.

Value

An object of class `GBIFLogin` containing the user's GBIF login data.

Examples

```
## Inputting user particulars
## Not run:
myLogin <- GBIFLoginManager(
  user = "theWoman",
  email = "ireneAdler@laScala.org",
  pwd = "sh3r"
)

## End(Not run)

## Not run:
## Can also be mined from your system environment
myLogin <- GBIFLoginManager(
  user = NULL,
```

```
  email = NULL, pwd = NULL
)

## End(Not run)
```

`getBIENpoints` *Download occurrence points from BIEN*

Description

Downloads occurrence points and useful related information for processing within other occCite functions

Usage

```
getBIENpoints(taxon)
```

Arguments

`taxon`	A single plant species or vector of plant species

Details

‘getBIENpoints‘ only returns all BIEN records, including non- native and cultivated occurrences.

Value

A list containing

1. a data frame of occurrence data;
2. a list containing: i notes on usage, ii bibtex citations, and iii acknowledgment information;
3. a data frame containing the raw results of a query to ‘BIEN::BIEN_occurrence_species()‘.

Examples

```
## Not run:
getBIENpoints(taxon = "Protea cynaroides")

## End(Not run)
```

`getGBIFpoints` *Download occurrences from GBIF*

Description

Downloads GBIF occurrence points and useful related information for processing within other occCite functions

Usage

```
getGBIFpoints(
  taxon,
  GBIFLogin = GBIFLogin,
  GBIFDownloadDirectory = NULL,
  checkPreviousGBIFDownload = T
)
```

Arguments

`taxon`	A string with a single species name
`GBIFLogin`	An object of class `GBIFLogin` to log in to GBIF to begin the download.
`GBIFDownloadDirectory`	An optional argument that specifies the local directory where GBIF downloads will be saved. If this is not specified, the downloads will be saved to your current working directory.
`checkPreviousGBIFDownload`	A logical operator specifying whether the user wishes to check their existing prepared downloads on the GBIF website.

Details

‘getGBIFpoints‘ only returns records from GBIF that have coordinates, aren’t flagged as having geospatial issues, and have an occurrence status flagged as "PRESENT".

Value

A list containing

1. a data frame of occurrence data;
2. GBIF search metadata;
3. a data frame containing the raw results of a query to ‘rgbif::occ_download_get()‘.

Examples

```
## Not run:
getGBIFpoints(
  taxon = "Gadus morhua",
  GBIFLogin = myGBIFLogin,
  GBIFDownloadDirectory = NULL
)

## End(Not run)
```

`myOccCiteObject` *Results of an occCite search for *Protea cynaroides**

Description

Results of an occCite search for *Protea cynaroides*

Usage

```
myOccCiteObject
```

Format

An ‘occCiteData‘ object with the following slots:

userQueryType What kind of query was made

userSpecTaxonomy A vector of taxonomic sources specified

cleanedTaxonomy A data frame with results of taxonomic cleanup

occSources A vector of which databases were queried (i.e. GBIF and BIEN)

occCiteSearchDate When the search was made

occResults A list of length 1 named "Protea cynaroides". Contains a list of length 2 with results from each database, GBIF and BIEN

Source

Global Biodiversity Information Facility, GBIF (`https://www.gbif.org/`) and Botanical Information and Ecology Network, BIEN (`https://bien.nceas.ucsb.edu/bien/`) data aggregators.

Examples

```
myOccCiteObject
```

`occCitation` *Occurrence Citations*

Description

Harvests citations for occurrence data

Usage

```
occCitation(x = NULL)
```

Arguments

`x`	An object of class `occCiteData`

Value

An object of class `occCiteCitation`. It is a named list of the same length as the number of species included in your `occCiteData` object. Each item in the list has citation information for occurrences.

Examples

```
## Not run:
data(myOccCiteObject)
myCitations <- occCitation(x = myOccCiteObject)

## End(Not run)
```

`occCiteCitation-class` *occCite Citation Class*

Description

A class for managing citations generated from occCite queries.

Fields

`occCitationResults` The results of performing `occCitation` on a `occCiteData` object, stored as a named list, each of the items named after a searched taxon and containing a data frame with occurrence information.

`occCiteData-class` *occCite Data Class*

Description

A class for managing metadata associated with occCite queries and data manipulation.

Slots

`userQueryType` A vector of type character specifying whether the user made their original taxonomic query based on a vector of taxon names or a phylogeny.

`userSpecTaxonomy` A vector of type character that presents a list of taxonomic sources for cleaning taxonomy of queries. This can be user-specified or default.

`cleanedTaxonomy` A data frame with containing input taxon names, the closest match according to `taxize::gnr_resolve`, and a list of taxonomic data sources that contain the matching name, generated by `studyTaxonList`.

`occSources` A vector of class "character" containing a list of occurrence data sources, generated when passing a `occCiteData` object through `occQuery`.

`occCiteSearchDate` The date on which the occurrence search query was conducted via occCite.

`occResults` The results of an `occQuery` search, stored as a named list, each of the items named after a searched taxon and containing a data frame with occurrence information.

`occCiteMap` *Generating a map of downloaded points*

Description

Makes maps for each individual species in an `occCiteData` object.

Usage

```
occCiteMap(
  occCiteData,
  species_map = "all",
  species_colors = NULL,
  ds_map = c("GBIF", "BIEN"),
  map_limit = 1000,
  awesomeMarkers = TRUE,
  cluster = FALSE
)
```

Arguments

`occCiteData`	An object of class `occCiteData` to map
`species_map`	Character; either the default "all" to map all species in `occCiteData`, or a subset of these specified as a character or character vector.
`species_colors`	Character; the default NULL will choose random colors from those available (see Details), or those specified by the user as a character or character vector (the number of colors must match the number of species mapped).
`ds_map`	Character; specifies which data service records will be mapped, with the default being GBIF, BIEN, and GBIF_BIEN (records with the same coordinates in both databases).
`map_limit`	Numeric; the number of points to map per species, set at a default of 1000 randomly selected records; users can specify a higher number, but be aware that leaflet can lag or crash when too many points are plotted.
`awesomeMarkers`	Logical; if ‘TRUE‘ (default), mapped points will be ‘awesomeMarkers‘ attributed with an icon for a globe for GBIF, a leaf for BIEN, or a database if records from both databases have the same coordinates; if ‘FALSE‘, mapped points will be leaflet ‘circleMarkers‘
`cluster`	Logical; if ‘TRUE‘ (default is ‘FALSE‘) turns on marker clustering, which does not preserve color differences between species

Details

When mapping using ‘awesomeMarkers‘ (default), the parameter species_colors must match those in a specified color library, currently: c("red", "lightred", "orange", "beige", "green", "lightgreen", "blue", "lightblue", "purple", "pink", "cadetblue", "white", "gray", "lightgray"). When ‘awesomeMarkers‘ is ‘FALSE‘ and species_colors are not specified, random colors from the ‘RColorBrewer‘ Set1 palette are used.

Value

A leaflet map

Examples

```
## Not run:
data(myOccCiteObject)
occCiteMap(myOccCiteObject, cluster = FALSE)

## End(Not run)
```

`occQuery` *Query from Taxon List*

Description

Takes rectified list of specimens from `studyTaxonList` and returns point data from `rgbif` with metadata.

Usage

```
occQuery(
  x = NULL,
  datasources = c("gbif", "bien"),
  GBIFLogin = NULL,
  GBIFDownloadDirectory = NULL,
  loadLocalGBIFDownload = F,
  checkPreviousGBIFDownload = T,
  options = NULL
)
```

Arguments

`x`	An object of class `occCiteData` (the results of a `studyTaxonList` search) OR a vector with a list of species names. Note: If the latter, taxonomic rectification uses NCBI taxonomies. If you want more control than this, use `studyTaxonList` to create a `occCiteData` object first.
`datasources`	A vector of occurrence data sources to search. This is currently limited to GBIF and BIEN, but may expand in the future.
`GBIFLogin`	An object of class `GBIFLogin` to log in to GBIF to begin the download.
`GBIFDownloadDirectory`	An optional argument that specifies the local directory where GBIF downloads will be saved. If this is not specified, the downloads will be saved to your current working directory.
`loadLocalGBIFDownload`	If `loadLocalGBIFDownload = T`, then occCite will load occurrences for the specified species that have been downloaded by the user and stored in the directory specified by `GBIFDownloadDirectory`.
`checkPreviousGBIFDownload`	If `loadLocalGBIFDownload = T`, occCite will check for previously-prepared GBIF downloads on the user's GBIF account. Setting this option to 'TRUE' can significantly speed up query time if the user has previously queried GBIF for the same taxa.
`options`	A vector of options to pass to `occ_download`.

Details

If you are querying GBIF, note that ‘occQuery()‘ only returns records from GBIF that have coordinates, aren’t flagged as having geospatial issues, and have an occurrence status flagged as "PRESENT".

Value

The object of class `occCiteData` supplied by the user as an argument, with occurrence data search results, as well as metadata on the occurrence sources queried.

Examples

```
## Not run:
## If you have already created a occCite object, and have not previously
## downloaded GBIF data.
occQuery(
  x = myOccCiteObject,
  datasources = c("gbif", "bien"),
  GBIFLogin = myLogin,
  GBIFDownloadDirectory = "./Desktop",
  loadLocalGBIFDownload = F
)

## If you don't have an occCite object yet
occQuery(
  x = c("Buteo buteo", "Protea cynaroides"),
  datasources = c("gbif", "bien"),
  GBIFLogin = myLogin,
  GBIFDownloadDirectory = "./Desktop",
  loadLocalGBIFDownload = F
)

## If you have previously downloaded occurrence data from GBIF
## and saved it in a folder called "GBIFDownloads".
occQuery(
  x = c("Buteo buteo", "Protea cynaroides"),
  datasources = c("gbif", "bien"),
  GBIFLogin = myLogin,
  GBIFDownloadDirectory = "./Desktop/GBIFDownloads",
  loadLocalGBIFDownload = T
)

## End(Not run)
```

`plot.occCiteData` *Plotting summary figures for occCite search results*

Description

Generates up to three different kinds of plots, with toggles determining whether plots should be done for individual species or aggregating all species–histogram by year of occurrence records, waffle::waffle plot of primary data sources, waffle::waffle plot of data aggregators.

Usage

```
## S3 method for class 'occCiteData'
plot(x, ...)
```

Arguments

`x`	An object of class `occCiteData` to map.
`...`	Additional arguments affecting how the formatted citation document is produced. ‘bySpecies‘: Logical; setting to ‘TRUE‘ generates the desired plots for each species. ‘plotTypes‘: The type of plot to be generated; "yearHistogram", "source", and/or "aggregator".

Value

A list containing the desired plots.

Examples

```
data(myOccCiteObject)
plot(
  x = myOccCiteObject, bySpecies = FALSE,
  plotTypes = c("yearHistogram", "source", "aggregator")
)
```

`prevGBIFdownload` *Download previously-prepared GBIF data sets*

Description

Searches the list of a user’s most recent 1000 downloads on the GBIF servers and returns the data set key for the most recently prepared download.

Usage

```
prevGBIFdownload(taxonKey, GBIFLogin)
```

Arguments

`taxonKey`	A taxon key as returned from ‘rgbif::name_suggest()‘.
`GBIFLogin`	An object of class `GBIFLogin` to log in to GBIF to begin the download.

Value

A GBIF download key, if one is available

Examples

```
## Not run:
GBIFLogin <- GBIFLoginManager(
  user = "theWoman",
  email = "ireneAdler@laScala.org",
  pwd = "sh3r"
)
taxKey <- rgbif::name_suggest(
  q = "Protea cynaroides",
  rank = "species"
)$key[1]
prevGBIFdownload(
  taxonKey = taxKey,
  GBIFLogin = myGBIFLogin
)

## End(Not run)
```

`print.occCiteCitation` *Print occCite citation object*

Description

Prints formatted citations for occurrences and main packages used (i.e. base, occCite, rgbif, and/or BIEN).

Usage

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

Arguments

`x`	An object of class `occCiteCitation`
`...`	Additional arguments affecting how the formatted citation document is produced

Value

A text string with formatted citations

Examples

```
# Print citations for all species together
data(myOccCiteObject)
print(myOccCiteObject)

# Print citations for each species individually
data(myOccCiteObject)
print(myOccCiteObject, bySpecies = TRUE)
```

`studyTaxonList` *Study Taxon List*

Description

Takes input phylogenies or vectors of taxon names, checks against taxonomic database, returns vector of cleaned taxonomic names (using `taxize::gnr_resolve()`) for use in spocc queries, as well as warnings if there are invalid names.

Usage

```
studyTaxonList(x = NULL, datasources = "GBIF Backbone Taxonomy")
```

Arguments

`x`	A phylogeny of class ’phylo’ or a vector of class ’character’ containing the names of taxa of interest
`datasources`	A vector of taxonomic data sources implemented in `taxize::gnr_resolve`. You can see the list using `taxize::gnr_datasources()`.

Value

An object of class `occCiteData` containing the type of inquiry the user has made –a phylogeny or a vector of names– and a data frame containing input taxa names, the closest match according to `taxize::gnr_resolve`, and a list of taxonomic data sources that contain the matching name.

Examples

```
## Inputting a vector of taxon names
studyTaxonList(
  x = c(
    "Buteo buteo",
    "Buteo buteo hartedi",
    "Buteo japonicus"
  ),
  datasources = c("National Center for Biotechnology Information")
)
```

```
## Inputting a phylogeny
phylogeny <- ape::read.nexus(
  system.file("extdata/Fish_12Tax_time_calibrated.tre",
    package = "occCite"
  )
)
phylogeny <- ape::extract.clade(phylogeny, 18)
studyTaxonList(
  x = phylogeny,
  datasources = c("GBIF Backbone Taxonomy")
)
```

`summary.occCiteData` *Summary for occCite data objects*

Description

Displays a summary of relevant stats about a query

Usage

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

Arguments

`object`	An object of class `occCiteData`
`...`	Additional arguments affecting the summary produced

Examples

```
data(myOccCiteObject)
summary(myOccCiteObject)
```

`taxonRectification` *Taxon Rectification*

Description

An function that takes an input taxonomic name, checks against taxonomic database, returns vector for use in database queries, as well as warnings if the name is invalid.

Usage

```
taxonRectification(taxName = NULL, datasources = NULL, skipTaxize = FALSE)
```

Arguments

`taxName`	A string that, ideally, is a taxonomic name
`datasources`	A vector of taxonomic data sources implemented in `taxize::gna_verifier()`. See the Global Names Verifier for more information.
`skipTaxize`	If `skipTaxize` = `TRUE`, occCite will skip taxonomic rectification using taxize. Setting this option to ‘TRUE‘ will result in a check for the `taxize` package before taxonomic rectification is attempted.

Value

A string with the closest match according to `taxize::gna_verifier()`, and a list of taxonomic data sources that contain the matching name.

Examples

```
# Inputting taxonomic name and specifying what taxonomic sources to search
taxonRectification(
  taxName = "Buteo buteo hartedi",
  datasources = "National Center for Biotechnology Information",
  skipTaxize = TRUE
)
```

Index