

# Package ‘occCite’

October 6, 2020

**Type** Package

**Title** Querying and Managing Large Biodiversity Occurrence Datasets

**Version** 0.3.0

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

**Maintainer** Hannah L. Owens <[hannah.owens@gmail.com](mailto: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 (>= 2)

**URL** <https://github.com/hannahlowens/occCite>

**BugReports** <https://github.com/hannahlowens/occCite/issues>

**Encoding** UTF-8

**LazyData** true

**Language** en-US

**Depends** R (>= 3.5.0)

**Suggests** rmarkdown, RColorBrewer, viridis, remotes

**Imports** RefManager, bib2df, BIEN, bit64, dplyr, ape, lubridate,  
methods, rgbif (>= 3.1), taxize, stringr, knitr, stats,  
leaflet, htmltools, ggplot2, rlang, magrittr, tidyr,  
RPostgreSQL, DBI, waffle

**VignetteBuilder** knitr

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-10-06 12:30:02 UTC

R topics documented:

BIENtableCleanup	2
GBIFLogin-class	3
GBIFLoginManager	3
getBIENpoints	4
getGBIFpoints	5
map.occCite	6
myOccCiteObject	7
occCitation	8
occCiteCitation-class	8
occCiteData-class	9
occQuery	9
prevGBIFdownload	11
print.occCiteCitation	12
studyTaxonList	12
sumFig.occCite	13
summary.occCiteData	14
tabulate.occResults	15
taxonRectification	15
<b>Index</b>	<b>17</b>

---

BIENtableCleanup	<i>BIEN Table Cleanup</i>
------------------	---------------------------

---

Description

Forces occurrence table columns to a uniform format. This is an internal function for use by [occQuery](#).

Usage

BIENtableCleanup(BIENtable)

Arguments

BIENtable      A data.frame resulting from [getBIENpoints](#).

Value

A data.frame tidied for compatibility with [getGBIFpoints](#) results.

Examples

```
temp <- getBIENpoints(taxon="Protea cynaroides", limit = NULL)
fitForUseTable <- BIENtableCleanup(temp$OccurrenceTable)
```

---

GBIFLogin-class	<i>GBIFLogin Data Class</i>
-----------------	-----------------------------

---

**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	<i>GBIF Login Manager</i>
------------------	---------------------------

---

**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 user-name.

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

myLogin <- GBIFLoginManager(user = "theWoman",
                             email = "ireneAdler@laScala.org",
                             pwd = "sh3r")

## Can also be mined from your system environment
myLogin <- GBIFLoginManager(user = NULL,
                             email = NULL, pwd = NULL)
```

---

getBIENpoints	<i>Download occurrence points from BIEN</i>
---------------	---------------------------------------------

---

## 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
-------	---------------------------------------------------

## 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

```
getBIENpoints(taxon="Protea cynaroides")
```

---

`getGBIFpoints`*Download occurrence points from GBIF*

---

### Description

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

### Usage

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

### Arguments

<code>taxon</code>	A single species
<code>GBIFLogin</code>	An object of class <a href="#">GBIFLogin</a> to log in to GBIF to begin the download.
<code>GBIFDownloadDirectory</code>	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.
<code>checkPreviousGBIFDownload</code>	A logical operator specifying whether the user wishes to check their existing prepared downloads on the GBIF website.

### 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

```
getGBIFpoints(taxon="Gadus morhua",  
              GBIFLogin = myGBIFLogin,  
              GBIFDownloadDirectory = NULL)
```

map.occCite

*Generating a map of downloaded points***Description**

Makes maps for each individual species in an [occCiteData](#) object.

**Usage**

```
map.occCite(
  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 <a href="#">occCiteData</a> to map
species_map	Character; either the default "all" to map all species in <a href="#">occCiteData</a> , 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**

```
data(myOccCiteObject)
map.occCite(myOccCiteObject, cluster = FALSE)
```

---

myOccCiteObject	<i>Results of an occCite search for *Protea cynaroides*</i>
-----------------	-------------------------------------------------------------

---

**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

**userSpecTaxonomicSources** 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

```
data(myOccCiteObject)
myCitations <- occCitation(x = myOccCiteObject)
```

---

 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	<i>occCite Data Class</i>
-------------------	---------------------------

---

### 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.

**userSpecTaxonomicSources** 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 [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.

---

occQuery	<i>Query from Taxon List</i>
----------	------------------------------

---

### 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 <code>occCiteData</code> (the results of a <code>studyTaxonList</code> 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 <code>studyTaxonList</code> to create a <code>occCiteData</code> 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 <code>GBIFLogin</code> 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 <code>loadLocalGBIFDownload = T</code> , then <code>occCite</code> will load occurrences for the specified species that have been downloaded by the user and stored in the directory specified by <code>GBIFDownloadDirectory</code> .
checkPreviousGBIFDownload	If <code>loadLocalGBIFDownload = T</code> , <code>occCite</code> 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 <code>occ_download</code> .

## 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

```
##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 a occCite object yet
occQuery(x = c("Buteo buteo", "Protea cynaroides"),
         datasources = c("gbif", "bien"),
         GBIFLogin = myLogin,
         GBIFOverwrite = T,
         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,
  GBIFoverwrite = T,
  GBIFDownloadDirectory = "./Desktop/GBIFDownloads",
  loadLocalGBIFDownload = T)
```

---

prevGBIFdownload	<i>Download previously-prepared GBIF data sets</i>
------------------	----------------------------------------------------

---

## 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 = "No key", GBIFLogin)
```

## Arguments

taxonKey	A taxon key as returned from 'rgbif::name_suggest()'.
GBIFLogin	An object of class <a href="#">GBIFLogin</a> to log in to GBIF to begin the download.

## Value

A GBIF download key, if one is available

## Examples

```
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)
```

---

```
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 <a href="#">occCiteCitation</a>
...	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 [taxonRectification](#)) for use in spocc queries, as well as warnings if there are invalid names.

### Usage

```
studyTaxonList(x = NULL, datasources = c("NCBI"))
```

**Arguments**

<code>x</code>	A phylogeny of class 'phylo' or a vector of class 'character' containing the names of taxa of interest
<code>datasources</code>	A vector of taxonomic data sources implemented in <a href="#">gnr_resolve</a> . You can see the list using <code>taxize::gnr_datasources()</code> .

**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 [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('NCBI'))

## 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('NCBI'))
```

---

sumFig.occCite

---

*Generating 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**

```
sumFig.occCite(
  occCiteData,
  bySpecies = FALSE,
  plotTypes = c("yearHistogram", "source", "aggregator")
)
```

**Arguments**

occCiteData	An object of class <code>occCiteData</code> to map.
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)
sumFig.occCite(myOccCiteObject, bySpecies = FALSE,
               plotType = c("yearHistogram", "source", "aggregator"))
```

---

summary.occCiteData	<i>Summary for occCite data objects</i>
---------------------	-----------------------------------------

---

**Description**

Displays a summary of relevant stats about a query

**Usage**

```
summary.occCiteData(object, ...)
```

**Arguments**

object	An object of class <code>occCiteData</code>
...	Additional arguments affecting the summary produced

**Examples**

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

---

tabulate.occResults	<i>Tabulate occurrence results</i>
---------------------	------------------------------------

---

**Description**

This is a helper function that tabulates ‘occCiteData’ objects for use by map.occCite and ‘sum-Fig.occCite’.

**Usage**

```
tabulate.occResults(x, sp.name)
```

**Arguments**

x	One species’ worth of results from an ‘occCiteData’ object
sp.name	A character string that is a species name from ‘x’

**Value**

A table that can be more easily mapped and used for summary plots.

**Examples**

```
data(myOccCiteObject)
tabulate.occResults(myOccCiteObject@occResults,
                    sp.name = "Protea cynaroides")
```

---

taxonRectification	<i>Taxon Rectification</i>
--------------------	----------------------------

---

**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)
```

**Arguments**

taxName	A string that, ideally, is a taxonomic name
datasources	A vector of taxonomic data sources implemented in <a href="#">gnr_resolve</a> . See the <a href="#">Global Names List</a> for more information.

**Value**

A string with the closest match according to [gnr\\_resolve](#), and a list of taxonomic data sources that contain the matching name.

**Examples**

```
#Inputting a taxonomic name and specifying what taxonomic sources you want to search
taxonRectification(taxName = "Buteo buteo hartedi", datasources = 'NCBI');
```



# Index

## \* **Internal**

tabulate.occResults, [15](#)

## \* **datasets**

myOccCiteObject, [7](#)

BIENtableCleanup, [2](#)

GBIFLogin, [3](#), [5](#), [10](#), [11](#)

GBIFLogin (GBIFLogin-class), [3](#)

GBIFLogin-class, [3](#)

GBIFLoginManager, [3](#)

getBIENpoints, [2](#), [4](#)

getGBIFpoints, [2](#), [5](#)

gnr\_resolve, [9](#), [13](#), [15](#), [16](#)

map.occCite, [6](#)

myOccCiteObject, [7](#)

occ\_download, [10](#)

occCitation, [8](#), [8](#)

occCiteCitation, [8](#), [12](#)

occCiteCitation  
(occCiteCitation-class), [8](#)

occCiteCitation-class, [8](#)

occCiteData, [6](#), [8–10](#), [13](#), [14](#)

occCiteData (occCiteData-class), [9](#)

occCiteData-class, [9](#)

occQuery, [2](#), [9](#), [9](#)

prevGBIFdownload, [11](#)

print.occCiteCitation, [12](#)

rgbif, [9](#)

studyTaxonList, [9](#), [10](#), [12](#)

sumFig.occCite, [13](#)

summary.occCiteData, [14](#)

tabulate.occResults, [15](#)

taxonRectification, [12](#), [15](#)