

# 1                   rgbif: R client for working with GBIF species occurrence data

2                                   Scott Chamberlain<sup>\*,a</sup>

3                                   <sup>a</sup>*University of California, Berkeley, CA, USA*

## 4   **Abstract**

- 5       1. xxx
- 6       2. xxx
- 7       3. xxx
- 8       4. xxxx

## 9   **Introduction**

10 Perhaps the most fundamental element in many fields of ecology is the individual. How many individuals  
11 of each species in a given location forms the basis for many sub-fields of ecology and evolution. Some  
12 research questions necessitate collecting new data, while others can easily take advantage of existing  
13 data. In fact, some ecology fields are built largely on existing data, e.g., macro-ecology (Brown, 1995;  
14 Beck et al., 2012).

15 Data on individuals, including which species, and where they're found, can be used for a large number of  
16 research questions. Biodiversity records have been used for a suite of other use cases: validating habitat  
17 suitability models with real occurrence data (Ficetola et al., 2014); ancestral range reconstruction  
18 (Ferretti et al., 2015; María Mendoza et al., 2015); development of invasive species watch lists (Faulkner  
19 et al., 2014); evaluate risk of invasive species spread (Febbraro et al., 2013); and effects of climate  
20 change on future biodiversity (Brown et al., 2015).

21 In addition to wide utility, this data is important for conservation. Biodiversity loss is one of the greatest  
22 challenges of our time (Pimm et al., 2014). Some have called this the sixth great mass extinction  
23 (Ceballos et al., 2015). Given this challenge there is a great need for data on specimen records, whether  
24 collected from live sightings in the field or specimens in museums.

25 There are many online services that collect and maintain specimen records. However, Global Biodiversity  
26 Information Facility (hereafter, GBIF, <http://www.gbif.org/>) is the largest collection of biodiversity

---

\*Corresponding author

Email address: `scott(at)ropensci.org` (Scott Chamberlain)

27 records globally, currently with 580 million records, 1.6 million taxa, 15,000 datasets from 770 publishers  
28 (figures collected on 2015-10-04). Many large biodiversity warehouses such as iNaturalist (<http://www.inaturalist.org/>),  
29 VertNet (<http://vertnet.org/>), and USGS's Biodiversity Information Serving Our  
30 Nation (BISON; <http://bison.usgs.ornl.gov/>) all feed into GBIF.

31 Herein, we describe the `rgbif` library (Chamberlain et al.) for working with GBIF data in the R  
32 programming environment (R Core Team, 2014). R is a widely used language in academia, and in  
33 non-profit and private sectors. Importantly, R makes it easy to do all of the steps of the research process,  
34 including data management, data manipulation and cleaning, statistics, and vizualization. Thus, an R  
35 client for getting GBIF data is a powerful tool to facilitate reproducible research.

## 36 **The `rgbif` package**

37 The `rgbif` package is completely written in R, uses an [MIT license](#) to maximize use everywhere. `rgbif`  
38 is developed publicly on GitHub at <https://github.com/ropensci/rgbif>, where development versions of  
39 the package can be installed, and bugs and feature requests reported. Stable versions of `rgbif` can be  
40 installed from [CRAN](#), the distribution network for R packages. `rgbif` is part of the rOpenSci project,  
41 a developer network making R software to facilitate reproducible research.

## 42 *Package interface*

43 `rgbif` is designed following the [GBIF Application Programming Interface](#), or API. The GBIF API has  
44 four major components: registry, species names, occurrences, and maps. We ignore maps in `rgbif` as  
45 it is concerned with generating maps primarily for web applications. `rgbif` has a suite of functions  
46 dealing with each of registry, species names, and occurrences - we'll go through each in turn describing  
47 design and example usage.

## 48 *Registry*

49 The GBIF registry API services are spread across four sets of functions:

- 50 • Datasets
- 51 • Installations
- 52 • Networks

- 53 • Nodes
- 54 • Organizations

55 The relationship between these five things is XXXXXX.

## 56 *Datasets*

57 Dataset functions include search, dataset metadata retrieval, and dataset metrics. Searching for datasets  
58 is an important part of the discovery process. One can search for datasets on the GBIF web portal.  
59 However, programmatic searching using this package is much more powerful. Identifying datasets  
60 appropriate for a research question is helpful as you can get metadata for each dataset, and track down  
61 dataset specific problems, if any.

62 The `dataset_search()` function is one way to search for datasets. Here, we search for the query term  
63 “oregon”, which finds any datasets that have terms matching that term.

```
res <- dataset_search(query = "oregon")
res$data$datasetTitle[1:10]
#> [1] "SDNHM Birds Collection"
#> [2] "CM Birds Collection"
#> [3] "condoncollection"
#> [4] "Taxonomy in Flux Checklist"
#> [5] "Wool carder bees of the genus Anthidium in the Western Hemisphere"
#> [6] "Bryophyte Collection - University of Washington Herbarium (WTU)"
#> [7] "University of British Columbia Herbarium (UBC) - Bryophytes Collection"
#> [8] "UWFC Ichthyology Collection"
#> [9] "Lichen Collection - University of Washington Herbarium (WTU)"
#> [10] "UWBM Mammalogy Collection"
```

64 Also, check out `datasets()` and `dataset_suggest()` for searching for datasets.

65 *Dataset metrics.* Dataset metrics are another useful way of digging in to figure out what datasets you  
66 may want to use. One drawback is that these data are only available for datasets of type *checklist*, but  
67 there are quite a lot of them (2472).

68 Here, we search for dataset metrics for a single dataset, with uuid `ec93a739-1681-4b04-b62f-3a687127a17f`,  
 69 a checklist of the ants (Hymenoptera: Formicidae) of the World.

```
res <- dataset_metrics(uuid='ec93a739-1681-4b04-b62f-3a687127a17f')
df <- data.frame(rank = names(res$countByRank),
                 count = unname(unlist(res$countByRank)))
knitr::kable(df)
```

rank	count
SPECIES	13710
SUBSPECIES	3234
GENUS	726
TRIBE	53
SUBFAMILY	20
FAMILY	2
KINGDOM	1
PHYLUM	1
CLASS	1
ORDER	1

70 *Networks, nodes, and installations*

71 Networks, nodes and installations are at a higher level of organization above datasets, but can be useful  
 72 if you want to explore data from given organizations. Here, we search for the first 10 GBIF networks,  
 73 returning just the title field.

```
networks(limit=10)$data$title
#> [1] "GBIF Backbone Sources"
#> [2] "Canadensys"
#> [3] "Southwest Collections of Arthropods Network (SCAN)"
#> [4] "VertNet"
#> [5] "Dryad"
#> [6] "GBIF Network"
```

```
#> [7] "The Knowledge Network for Biocomplexity (KNB) "
#> [8] "Online Zoological Collections of Australian Museums (OZCAM)"
#> [9] "Catalogue of Life"
#> [10] "Ocean Biogeographic Information System (OBIS)"
```

74 *Species*

75 *Occurrences*

76 GBIF provides two ways to get occurrence data: through the `/occurrence/search` route (see  
 77 `occ_search`), or via the `/occurrence/download` route (many functions, see below). `occ_search()` is  
 78 the main function for the search route, and is more appropriate for smaller data, while `occ_download*()`  
 79 functions are more appropriate for larger data requests.

80 Large is of course a subjective term. When you hit a “large dataset” will depend primarily on the size  
 81 of the your data request. GBIF imposes for any given search a limit of 200,000 records in the search  
 82 service, after which point you can’t download any more records for that search. However, you can  
 83 download more records for different searches.

84 We think the search service is still quite useful for many people even given the 200,000 limit. For those  
 85 that need more data, we have created a similar interface in the `download_*()` functions, that should be  
 86 easy to use. Users should take note that using the download service has a few extra steps to get data  
 87 into R, but is straight-forward.

88 *Download API*

89 The download API syntax is similar to the occurrence search API in that the same parameters are used,  
 90 but the way in which the query is defined is different. For example, in the download API you can do  
 91 greater than searches (i.e., `latitude > 50`), whereas you can not do that in the occurrence search API.  
 92 Thus, we can’t make the query interface exactly the same for both search and download functions.

93 Using the download service can be as few as three steps: 1) Request data via a search; 2) Download  
 94 data; 3) Import data into R.

95 Request data download given a query. Here, we search for the taxon key 3119195, which is the key for  
 96 *Helianthus annuus* (<http://www.gbif.org/species/3119195>).

```
occ_download('taxonKey = 3119195')
#> <<gbif download>>
#> Username: xxxx
#> E-mail: xxxx
#> Download key: 0000840-150615163101818
```

97 You can check on when the download is ready using the functions `occ_download_list()` and  
 98 `occ_download_meta()`. When it's ready use `occ_download_get()` to download the dataset to your  
 99 computer.

```
(res <- occ_download_get("0000840-150615163101818", overwrite = TRUE))
#> <<gbif downloaded get>>
#> Path: ./0000840-150615163101818.zip
#> File size: 3.19 MB
```

100 What's printed out above is a very brief summary of what was downloaded, the path to the file, and its  
 101 size (in human readable form).  
 102 Next, read the data in to R using the function `occ_download_import()`.

```
library("dplyr")
dat <- occ_download_import(res)
dat %>%
  select(gbifID, decimalLatitude, decimalLongitude)
#>      gbifID decimalLatitude decimalLongitude
#> 1  657590544             NA              NA
#> 2  657679551             NA              NA
#> 3  657791316       37.70805       -118.4162
#> 4  658180562             NA              NA
#> 5  441881672             NA              NA
#> 6  911596181             NA              NA
#> 7   56454601             NA              NA
#> 8  657848913             NA              NA
```

```
#> 9 658187373 NA NA
#> 10 658279212 38.95917 -106.9892
#> .. ... ..
```

103 *Downloaded data format.* The downloaded dataset from GBIF is actually a Darwin Core Archive  
 104 (DwC-A), an internationally recognized biodiversity informatics standard (<http://rs.tdwg.org/dwc/>).  
 105 The DwC-A downloaded is a compressed folder with a number of files, including metadata, citations for  
 106 each of the datasets included in the download, and the data itself, in separate files for each dataset as  
 107 well as one single `.txt` file. In `occ_download_import()`, we simply fetch data from the `.txt` file. If you  
 108 want to dig into the metadata, citations, etc., it is easily accessible from the folder on your computer.

#### 109 *Search API*

110 The search API follows the GBIF API and is broken down into the following functions:

- 111 • `occ_count()`
- 112 • `occ_search()`
- 113 • `occ_get()`
- 114 • `occ_metadata()`

115 The main search work-horse is `occ_search()`. This function allows very flexible search definitions. In  
 116 addition, this function does paging internally, making it such that the user does not have worry about  
 117 the 300 records per request limit - but of course we can't go over the 200,000 maximum limit.

118 ...

119 *Cleaning data.* GBIF provides optional data issues with each occurrence record. These issues fall into  
 120 many different pre-defined classes, covering issues with taxonomic names, geographic data, and more  
 121 (see `occ_issues_lookup()` to find out more information on GBIF issues; and the same data on [GBIF's](#)  
 122 [development site](#)).

123 `occ_issues()` provides a way to easily filter data downloaded via `occ_search()` based on GBIF issues.

```

out <- occ_search(issue='DEPTH_UNLIKELY', limit = 500)
NROW(out)
#> [1] 4
out %>% occ_issues(-cudc) %>% .$data %>% NROW
#> [1] 2

```

124 *Use cases*

125 *A*

126 *B*

127 *C*

128 **Conclusions and future directions**

- 129 • pt 1
- 130 • pt 2
- 131 • pt 3
- 132 • pt 4

133 *Acknowledgements*

134 This project was supported in part by the Alfred P Sloan Foundation (Grant 2013-6-22).

135 *Data Accessibility*

136 All scripts and data used in this paper can be found in the permanent data archive Zenodo under  
 137 the digital object identifier (DOI). This DOI corresponds to a snapshot of the GitHub repository at  
 138 [github.com/sckott/msrgbif](https://github.com/sckott/msrgbif). Software can be found at [github.com/ropensci/rgbif](https://github.com/ropensci/rgbif), under the open and  
 139 permissive MIT license.

140 **References**

141 Beck J., Ballesteros-Mejia L., Buchmann CM., Dengler J., Fritz SA., Gruber B., Hof C., Jansen  
 142 F., Knapp S., Kreft H., Schneider A-K., Winter M., Dormann CF. 2012. Whats on the horizon for  
 143 macroecology? *Ecography* 35:673–683.



144 Brown JH. 1995. *Macroecology*. University of Chicago Press.

145 Brown KA., Parks KE., Bethell CA., Johnson SE., Mulligan M. 2015. Predicting plant diversity patterns  
 146 in madagascar: Understanding the effects of climate and land cover change in a biodiversity hotspot.  
 147 *PLOS ONE* 10:e0122721.

148 Ceballos G., Ehrlich PR., Barnosky AD., Garcia A., Pringle RM., Palmer TM. 2015. Accelerated  
 149 modern human-induced species losses: Entering the sixth mass extinction. *Science Advances* 1:e1400253–  
 150 e1400253.

151 Chamberlain S., Ram K., Barve V., Mcglinn D. *Rgbif: Interface to the global 'biodiversity' information*  
 152 *facility 'aPI'*.

153 Faulkner KT., Robertson MP., Rouget M., Wilson JR. 2014. A simple, rapid methodology for developing  
 154 invasive species watch lists. *Biological Conservation* 179:25–32.

155 Febbraro MD., Lurz PWW., Genovesi P., Maiorano L., Girardello M., Bertolino S. 2013. The use of  
 156 climatic niches in screening procedures for introduced species to evaluate risk of spread: A case with  
 157 the american eastern grey squirrel. *PLoS ONE* 8:e66559.

158 Ferretti F., Verd GM., Seret B., Šprem JS., Micheli F. 2015. Falling through the cracks: The fading  
 159 history of a large iconic predator. *Fish and Fisheries*:n/a–n/a.

160 Ficetola GF., Rondinini C., Bonardi A., Baisero D., Padoa-Schioppa E. 2014. Habitat availability for  
 161 amphibians and extinction threat: A global analysis. *Diversity and Distributions* 21:302–311.

162 María Mendoza., Ospina OE., Cárdenas-Henao H., García-R JC. 2015. A likelihood inference of  
 163 historical biogeography in the world's most diverse terrestrial vertebrate genus: Diversification of  
 164 direct-developing frogs (craugastoridae: Pristimantis) across the neotropics. *Molecular Phylogenetics*  
 165 *and Evolution* 85:50–58.

166 Pimm SL., Jenkins CN., Abell R., Brooks TM., Gittleman JL., Joppa LN., Raven PH., Roberts CM.,  
 167 Sexton JO. 2014. The biodiversity of species and their rates of extinction, distribution, and protection.  
 168 *Science* 344:1246752–1246752.

169 R Core Team. 2014. *R: A language and environment for statistical computing*. Vienna, Austria: R  
 170 Foundation for Statistical Computing.