

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

2 Scott Chamberlain^{*,a}

3 ^a*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.](http://www.inaturalist.org/)
29 [inaturalist.org/](http://www.inaturalist.org/)), 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
44 has four major components: registry, taxonomic names, occurrences, and maps. We ignore maps in
45 `rgbif` as it is concerned with generating maps primarily for web applications. `rgbif` has a suite of
46 functions dealing with each of registry, taxonomic names, and occurrences - we'll go through each in
47 turn describing 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

- Nodes
- Organizations

Datasets are owned by organizations. Organizations are endorsed by nodes to share datasets with GBIF. Datasets are published through institutions, which may be hosted at another organization. A network is a group of datasets (managed by GBIF).

Datasets

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

The `dataset_search()` function is one way to search for datasets. Here, we search for the query term “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"
```

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

67 *Dataset metrics.* Dataset metrics are another useful way of digging in to figure out what datasets you
 68 may want to use. One drawback is that these data are only available for datasets of type *checklist*, but
 69 there are quite a lot of them (2474).

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

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

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

72 *Networks, nodes, and installations*

73 Networks, nodes and installations are at a higher level of organization above datasets, but can be useful
 74 if you want to explore data from given organizations. Here, we search for the first 10 GBIF networks,
 75 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)"
```

76 *Taxonomic names*

77 The GBIF taxonomic names API services are spread across five functions:

- 78 • Search GBIF name backbone - `name_backbone()`
- 79 • Search across all checklists - `name_lookup()`
- 80 • Quick name lookup - `name_suggest()`
- 81 • Name usage of a name according to a checklist - `name_usage()`
- 82 • GBIF name parser - `parsenames()`

83 The goal of these name functions is often to settle on a taxonomic name known to GBIF's database.
 84 This serves two purposes: 1) when referring to a taxonomic name, you can point to a URI on the
 85 internet, and 2) you can search for metadata on that species, and occurrences of that species in GBIF.
 86 Taxonomic names are particularly tricky. Many different organizations have their own unique codes for
 87 the same taxonomic names, and some taxonomic groups have preferred sources for the definitive names
 88 for that group.

89 When searching for occurrences (see below) you can search by taxonomic name (and other filters, e.g.,
 90 taxonomic rank), but you're probably better off figuring out the taxonomic key in the GBIF backbone
 91 taxonomy, and using that to search for occurrences. The `taxonkey` parameter in the GBIF occurrences
 92 API expects a GBIF backbone taxon key.

93 *GBIF Backbone*

94 The GBIF backbone taxonomy is used in GBIF to have a consistent way to refer to taxonomic
 95 names throughout their services. The backbone has 4410899 unique names and 2497114 species

96 names. The backbone taxonomy is also a dataset with key `d7dddbf4-2cf0-4f39-9b2a-bb099caae36c`
97 (<http://www.gbif.org/dataset/d7dddbf4-2cf0-4f39-9b2a-bb099caae36c>).

98 We can search the backbone taxonomy with the function `name_backbone()`. Here, we're searching for
99 the name *Poa*, restricting to genera, and the family *Poaceae*.

```
res <- name_backbone(name='Poa', rank='genus', family='Poaceae')
res[c('usageKey', 'kingdom')]
#> $usageKey
#> [1] 2704173
#>
#> $kingdom
#> [1] "Plantae"
```

100 Name searching

101 One of the quickest ways to search for names is using `name_suggest()`, which does a very quick search
102 and returns minimal data. Here, we're searching for the query term *Pum*, and we get back a bunch of
103 names:

```
name_suggest(q='Pum', limit = 6)
```

key	canonicalName	rank
3269133	Pumilus	GENUS
4407849	Pumilinura	GENUS
4323990	Pumiliopsis	GENUS
4324083	Pumiliopes	GENUS
4161281	Pumilea	GENUS
4312370	Pumilopagurus	GENUS

104 Occurrences

105 GBIF provides two ways to get occurrence data: through the `/occurrence/search` route (see
106 `occ_search`), or via the `/occurrence/download` route (many functions, see below). `occ_search()` is

107 the main function for the search route, and is more appropriate for smaller data, while `occ_download*()`
108 functions are more appropriate for larger data requests.

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

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

117 *Download API*

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

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

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

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

126 You can check on when the download is ready using the functions `occ_download_list()` and
127 `occ_download_meta()`. When it’s ready use `occ_download_get()` to download the dataset to your
128 computer.

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

129 What's printed out above is a very brief summary of what was downloaded, the path to the file, and its
130 size (in human readable form).

131 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  725767384      61.01005      24.41740
#> 2  725767447      59.82923      23.13550
#> 3  725767450      60.38505      25.17449
#> 4  725767513      68.37648      23.51963
#> 5  725767546      67.19203      24.85820
#> 6  725767579      60.21607      24.67412
#> 7  725767609      66.49260      25.70471
#> 8  725767645      61.36634      24.76218
#> 9  725767678      62.29174      27.96500
#> 10 725767681      60.28615      22.38489
#> ..      ...      ...
```

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

138 *Search API*

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

- 140 • Get a single numeric count of occurrences - `occ_count()`
- 141 • Search for occurrences - `occ_search()`
- 142 • Get occurrences by occurrence identifier - `occ_get()`
- 143 • Get occurrence metadata - `occ_metadata()`

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

147 ...

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

152 `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
```

153 *Use cases*

154 *A*

155 *B*

156 *C*

157 **Conclusions and future directions**

- 158 • pt 1

- pt 2
- pt 3
- pt 4

Acknowledgements

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

Data Accessibility

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

References

- Beck J., Ballesteros-Mejia L., Buchmann CM., Dengler J., Fritz SA., Gruber B., Hof C., Jansen F., Knapp S., Kreft H., Schneider A-K., Winter M., Dormann CF. 2012. Whats on the horizon for macroecology? *Ecography* 35:673–683.
- Brown JH. 1995. *Macroecology*. University of Chicago Press.
- Brown KA., Parks KE., Bethell CA., Johnson SE., Mulligan M. 2015. Predicting plant diversity patterns in madagascar: Understanding the effects of climate and land cover change in a biodiversity hotspot. *PLOS ONE* 10:e0122721.
- Ceballos G., Ehrlich PR., Barnosky AD., Garcia A., Pringle RM., Palmer TM. 2015. Accelerated modern human-induced species losses: Entering the sixth mass extinction. *Science Advances* 1:e1400253–e1400253.
- Chamberlain S., Ram K., Barve V., Mcglinn D. *Rgbif: Interface to the global 'biodiversity' information facility 'aPI'*.
- Faulkner KT., Robertson MP., Rouget M., Wilson JR. 2014. A simple, rapid methodology for developing invasive species watch lists. *Biological Conservation* 179:25–32.

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

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

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

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

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

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