GBIF Occurrence Data: from search to map

Patrícia Singh

2025-09-24

title: “GBIF Occurrence Data: from search to map” author: “Patrícia Singh” date: “2025-09-24” format: html: toc: true toc-depth: 3 number-sections: true code-fold: show code-tools: true pdf: default docx: default editor: visual jupyter: r execute: echo: true warning: false message: false

> This Quarto document (`.qmd`) is ready to knit to **HTML/PDF/DOCX** and to publish on a website (Quarto Pub or GitHub Pages). It cleans up and structures your commented script into a reproducible workflow.

1. Packages & setup

```
# Install (first time) and load required packages # install.packages(c("rgbif", "dplyr", "map
```

What we’ll do

1.
2. Quick counts from GBIF (occurrences & observations).
3.
4. Explore one species (*Sphagnum fuscum*).
5.
6. Handle GBIF’s 10k download limit: `occ_search()` vs `occ_download()`.
7.
8. Save the file and import the full dataset.
9.
10. Make a simple map.

11.

Tip: Occurrence vs Observation

-

- **Occurrence** = any presence record (specimens, fossils, observations, eDNA…).

-

- **Observation** = specifically field/machine observations (e.g. iNaturalist, bird checklists, camera traps).

  :::

-

2. Quick counts from GBIF

```
# All GBIF occurrences (all bases of record) occ_count()  # Observations only occ_count(bas
```

3. Example species: *Sphagnum fuscum*

```
# Check accepted name and key (synonyms) name_suggest(q = "Sphagnum fuscum", rank = "specie
```

Heads-up: `occ_search()` limits

`occ_search()` will **not** return more than **100,000** records in one call. If there are more, filter by time (`year`), geography (`decimalLatitude`/`decimalLongitude`), or tile the area and loop. For truly large pulls, use `occ_download()` below.

Filter to human observations only

```
occ_search(   scientificName = "Sphagnum fuscum",   hasCoordinate  = TRUE,   basisOfRecord
```

Pull up to 10k records with `occ_search()`

```
# Up to 10,000 records directly into R sph_occ <- occ_search(   scientificName = "Sphagnum
```

4. Large downloads with `occ_download()` (recommended)

For complete datasets (beyond 10k), request a **server-side** GBIF download and then fetch it.

4.1. Create `~/.Renviron` with GBIF credentials (one-time)

```
# Where is your home directory (Windows example)? path.expand("~") # Create a text file nam
```

Windows gotcha

If `Sys.getenv()` returns empty strings, your file is probably `*.Renviron.txt*`.

-
- In **File Explorer** → **View** → **Show** → **File name extensions**, then rename to **`.Renviron`** (no `.txt`).

  :::
-

4.2. Submit the download request

```
# Get the GBIF taxonKey for Sphagnum fuscum key <- name_backbone(name = "Sphagnum fusc
```

4.3. Wait until it's ready, then fetch & import

```
# Block until GBIF marks it as finished occ_download_wait(req)  # Download the zip loc
```

Typical metadata echoed after `occ_download_wait()` include **Status**, **DOI**, **Download key**, **Created/Modified** timestamps, and **Total records**. Keep the DOI for citation.

5. Quick mapping

```
# Simple world map then points map("world") points(dat$decimalLongitude, dat$decimalLa
```

To show the **entire world**, just use `map("world")` without `xlim/ylim`. To focus on Europe, set `xlim/ylim` as above.

6. Reproducibility

```
sessionInfo()
```

7. Publishing options

-
- **Quarto Pub**: `quarto publish` directly from RStudio/VS Code.
-
- **GitHub Pages**: put this file in a Quarto project (`_quarto.yml`), render to `docs/`, and enable Pages.
-
- **Static upload**: Knit to HTML and upload the single HTML file to your site.
-

Minimal `_quarto.yml` for a website

```
project:   type: website   output-dir: docs  website:   title: "Ecological Forecasting
```

> Save this file next to your `.qmd`, render with `quarto render`, then push to GitHub and enable **GitHub Pages** (branch `main`, folder `/docs`).