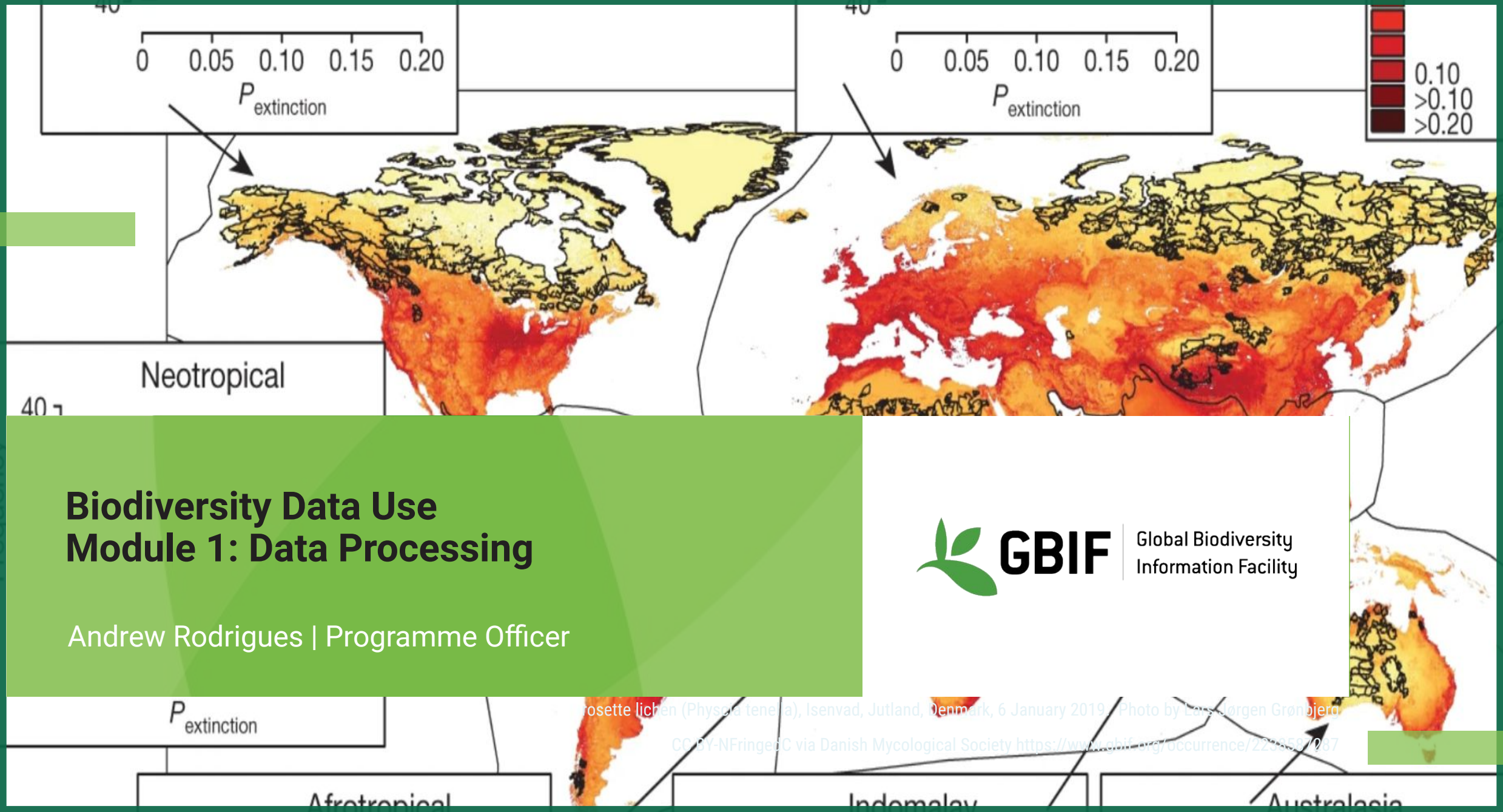
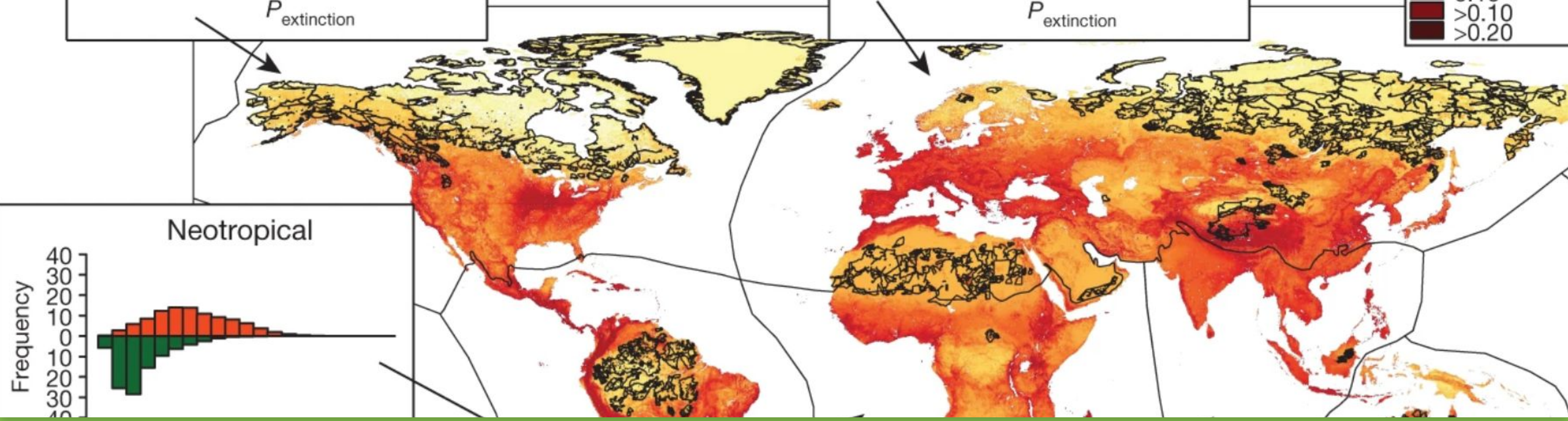


Biodiversity Data Use
Module 1: Data Processing

Andrew Rodrigues | Programme Officer

GBIF Global Biodiversity Information Facility

rosette lichen (Physcia tenella), Isenvad, Jutland, Denmark, 6 January 2019. Photo by Lars Jørgen Grønbjerg
CC BY-NFringeEC via Danish Mycological Society https://www.gbif.org/occurrence/2208507037



COURSE OUTLINE

Part 1 - Navigating www.gbif.org
Part 2- Common Data Quality Issues
Part 3- The API
Part 4 - Using R

Resources: **https://docs.gbif.org/course-data-use/en/key-documentation.html**

Community Forum: https://discourse.gbif.org/g/BID_DataUse

Di Marco M, Ferrier S, Harwood TD, Hoskins AJ and Watson JEM (2019) Wilderness areas halve the extinction risk of terrestrial biodiversity. Nature. Springer Science and Business Media LLC 573(7775): 582–585. Available at: https://doi.org/10.1038/s41586-019-1567-7

GBIF

Trainers and Mentors

Trainers

Andrew Rodrigues – GBIF Secretariat, Programme Office for Participation and Engagement

John Waller - GBIF Secretariat, Data Analyst

Mentors

Anabela Plos

Museo Argentino de Ciencias Naturales Bernardino Rivadavia

Arman Pili

Monash Univerity

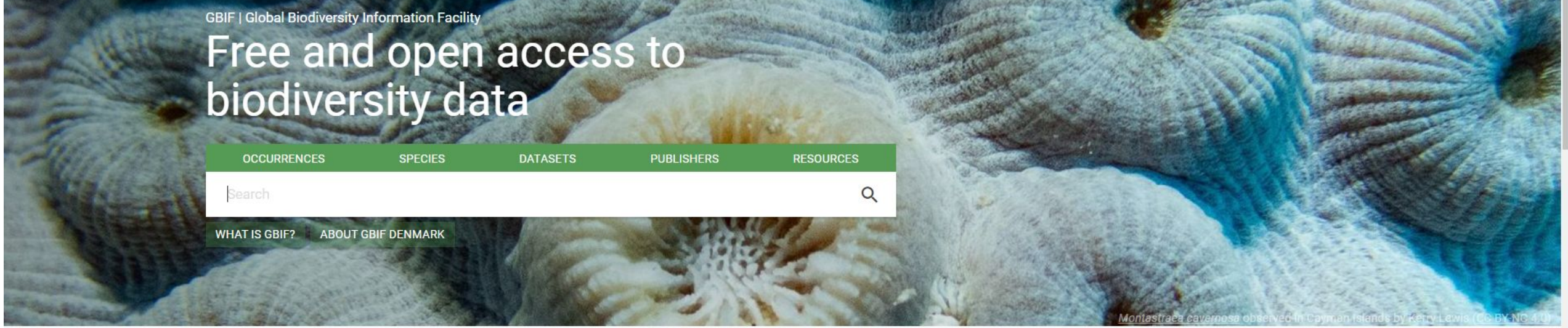
Leonardo Buitrago

BID Caribbean Regional Support Contractor

Vijay Barve

University of Florida

GBIF



Occurrence records	Datasets	Publishing institutions	Peer-reviewed papers using data
1,676,825,999	57,757	1,665	5,703

WHY DO WE PROCESS DATA?

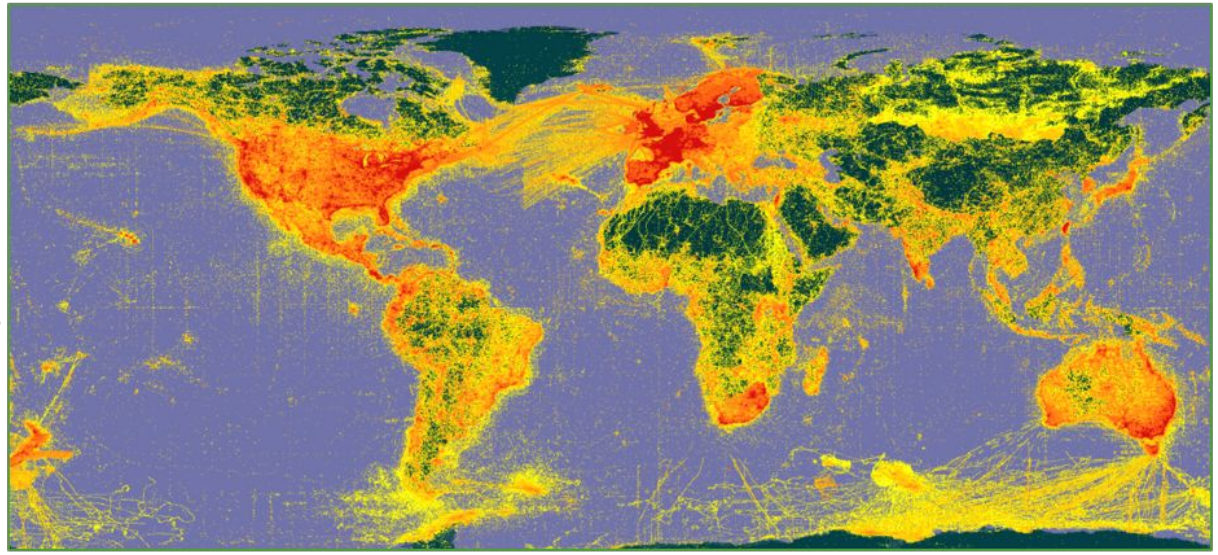
We want to make our download fit for our own purposes

1. Remove erroneous data e.g. outliers
2. Ensure sufficient level of precision in the data for our purpose

GBIF

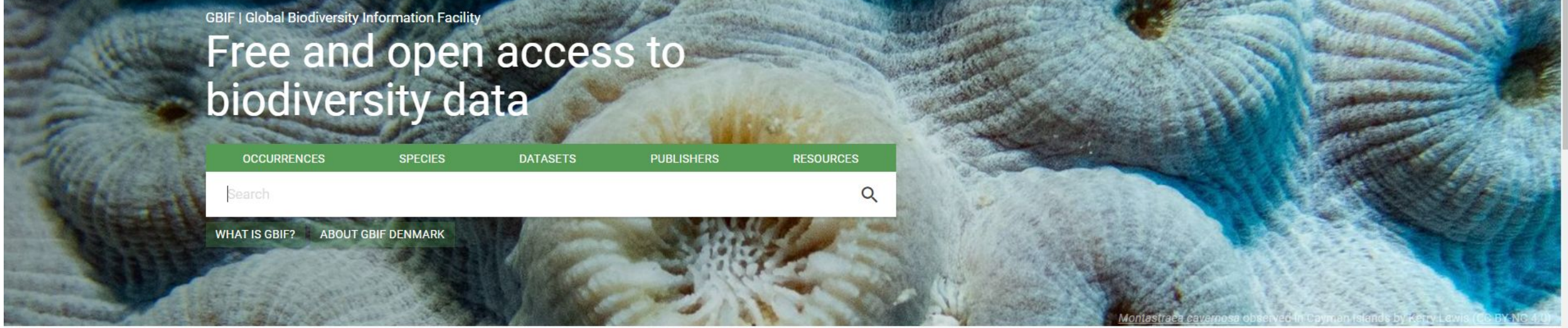
A WINDOW ON EVIDENCE ABOUT WHERE SPECIES HAVE LIVED, AND WHEN

- Observations
- Digitized specimens
- Literature
- Data publishing and indexing
- Common standards (DwC)
- Remote-sensing
- Environmental DNA



- Data discovery and use

GBIF



Occurrence records	Datasets	Publishing institutions	Peer-reviewed papers using data
1,676,825,999	57,757	1,665	5,703

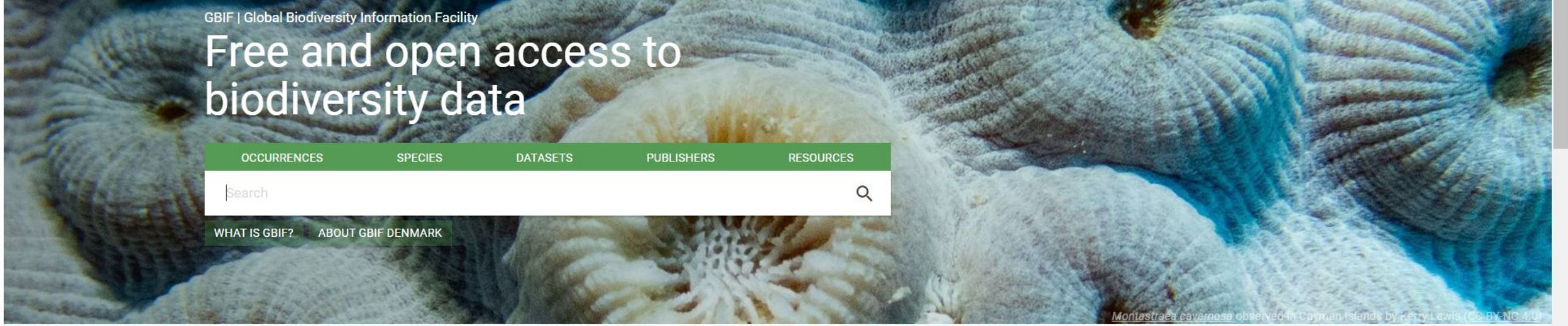
WHY DO WE PROCESS DATA?

Each time you process a dataset for use you will have to consider

1. Requirements of your analysis
2. Balance between data quality and the robustness of your analysis

This may be an iterative process

GBIF



Occurrence records	Datasets	Publishing institutions	Peer-reviewed papers using data
1,676,825,999	57,757	1,665	5,703

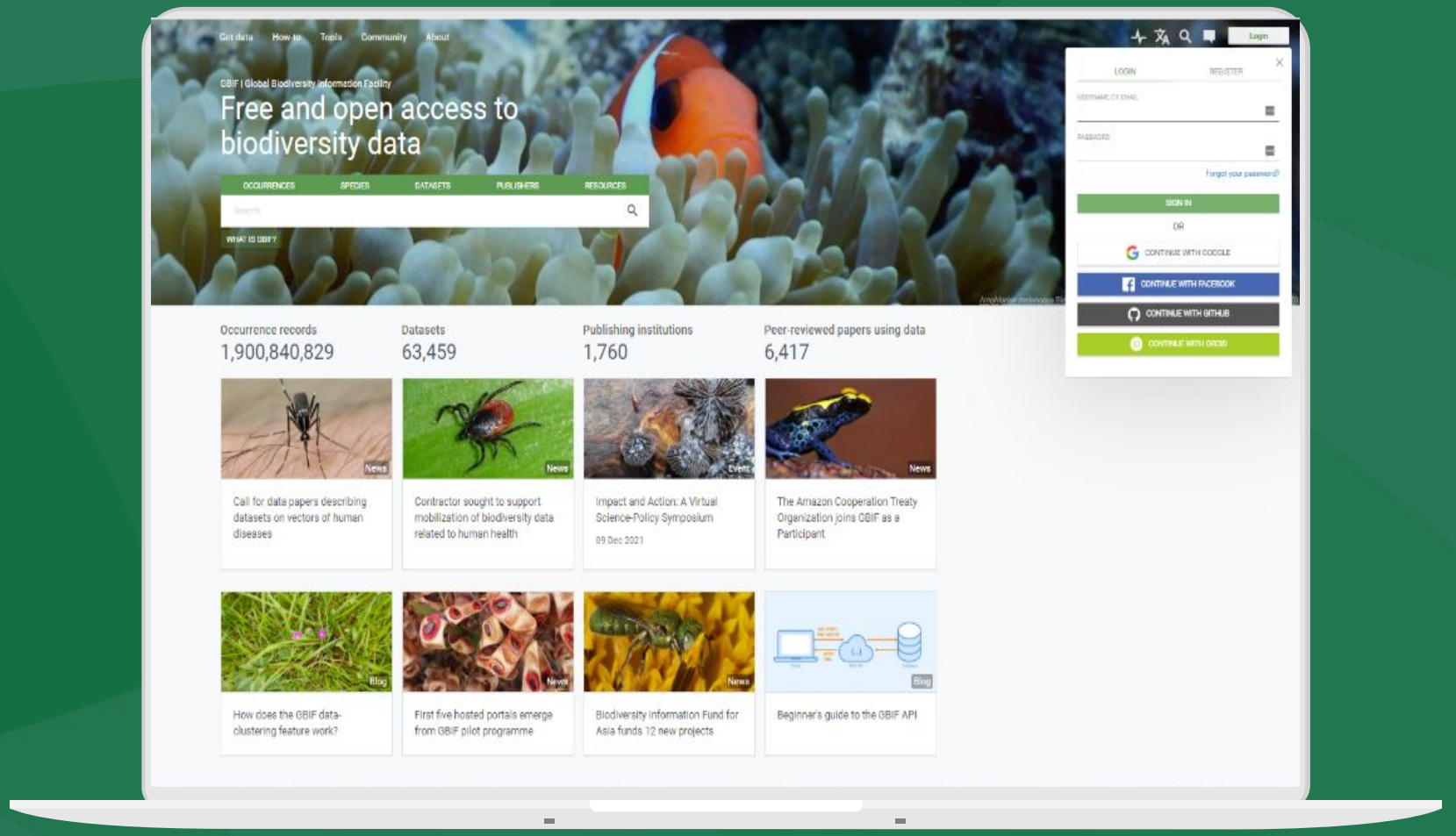
GOLDEN RULES OF GBIF-MEDIATED DATA USE

1. Must have an account on www.gbif.org
2. Must agree to the Data User Agreement - https://www.gbif.org/terms/data-user
3. Document how you process your data
4. Correctly cite the data you use
5. Deposit used data in a public repository

GBIF

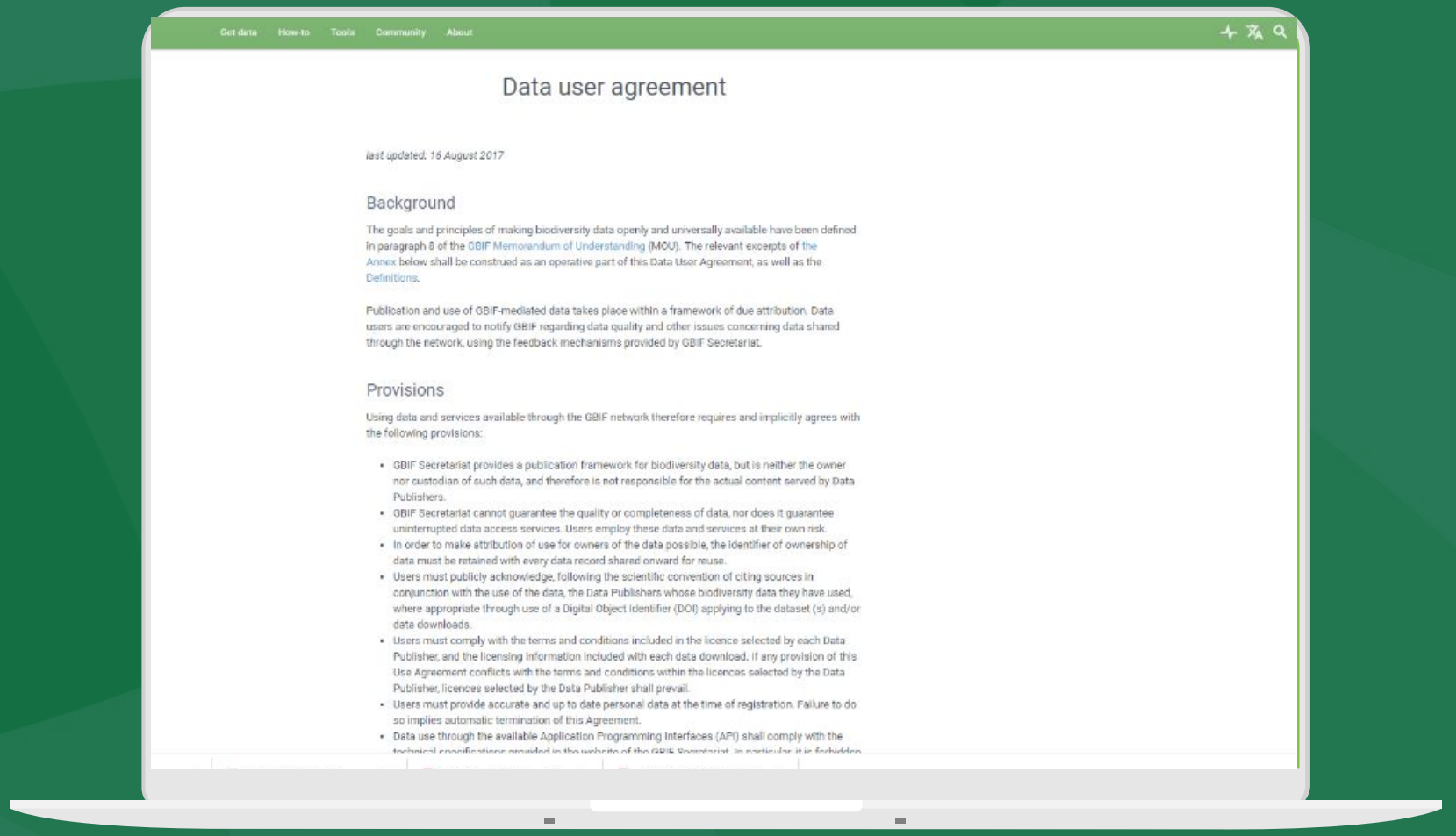
GOLDEN RULES OF GBIF-MEDIATED DATA USE

- Must have an account on www.gbif.org



GOLDEN RULES OF GBIF-MEDIATED DATA USE

- Must have an account on www.gbif.org
- Must agree to the Data User Agreement - https://www.gbif.org/terms/data-user
  - Non-binding
  - Sets out guiding principles of data use including citation of data



GOLDEN RULES OF GBIF-MEDIATED DATA USE

- Must have an account on www.gbif.org
- Must agree to the Data User Agreement - https://www.gbif.org/terms/data-user
- Document how you process your data

Step 1 Download occurrence records for species *x* with an associated DOI

Step 2 Remove all records from outside its native range

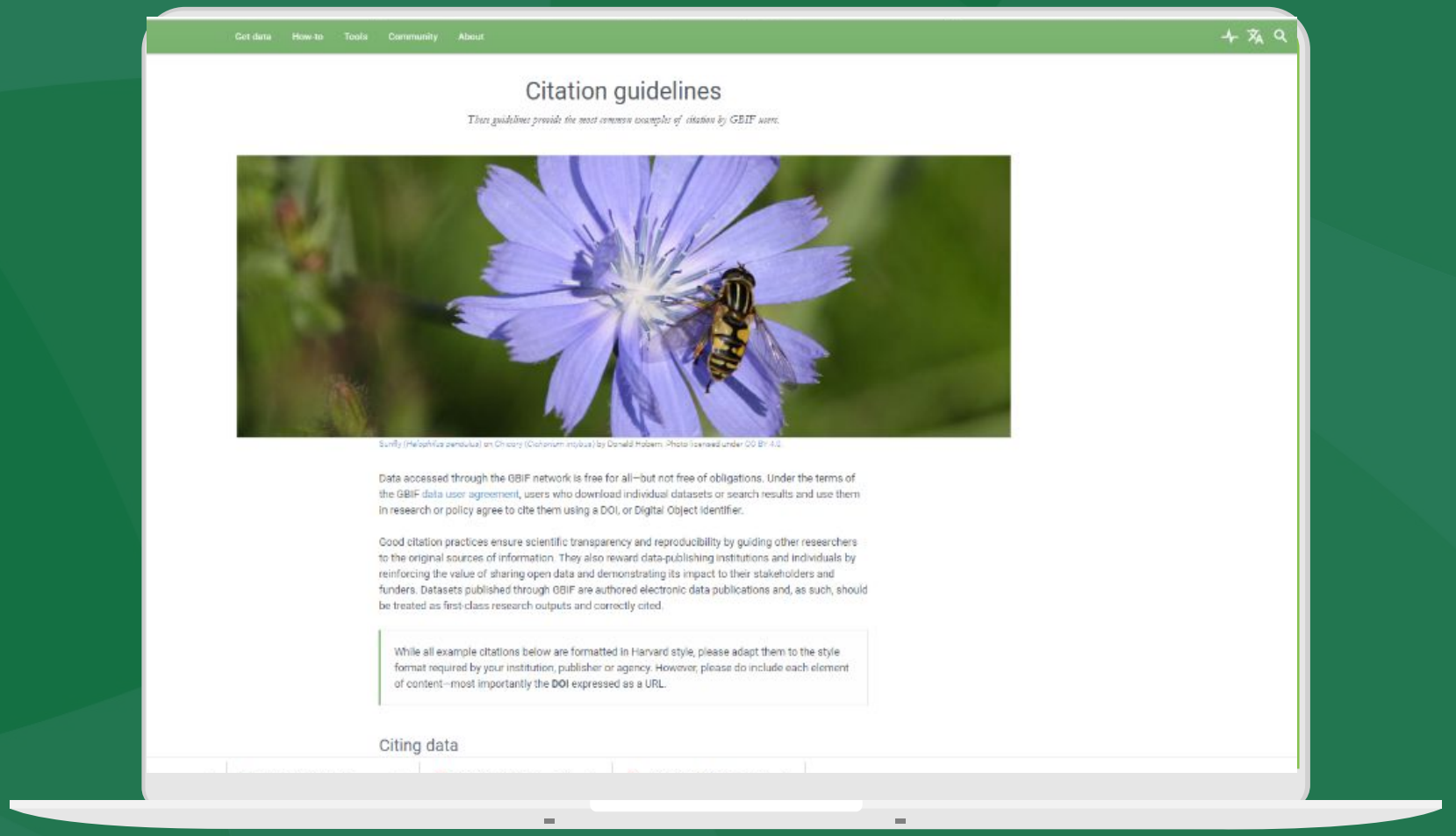
Step 3 Remove all records collected before 1950

Step 4 Final clean dataset

GBIF

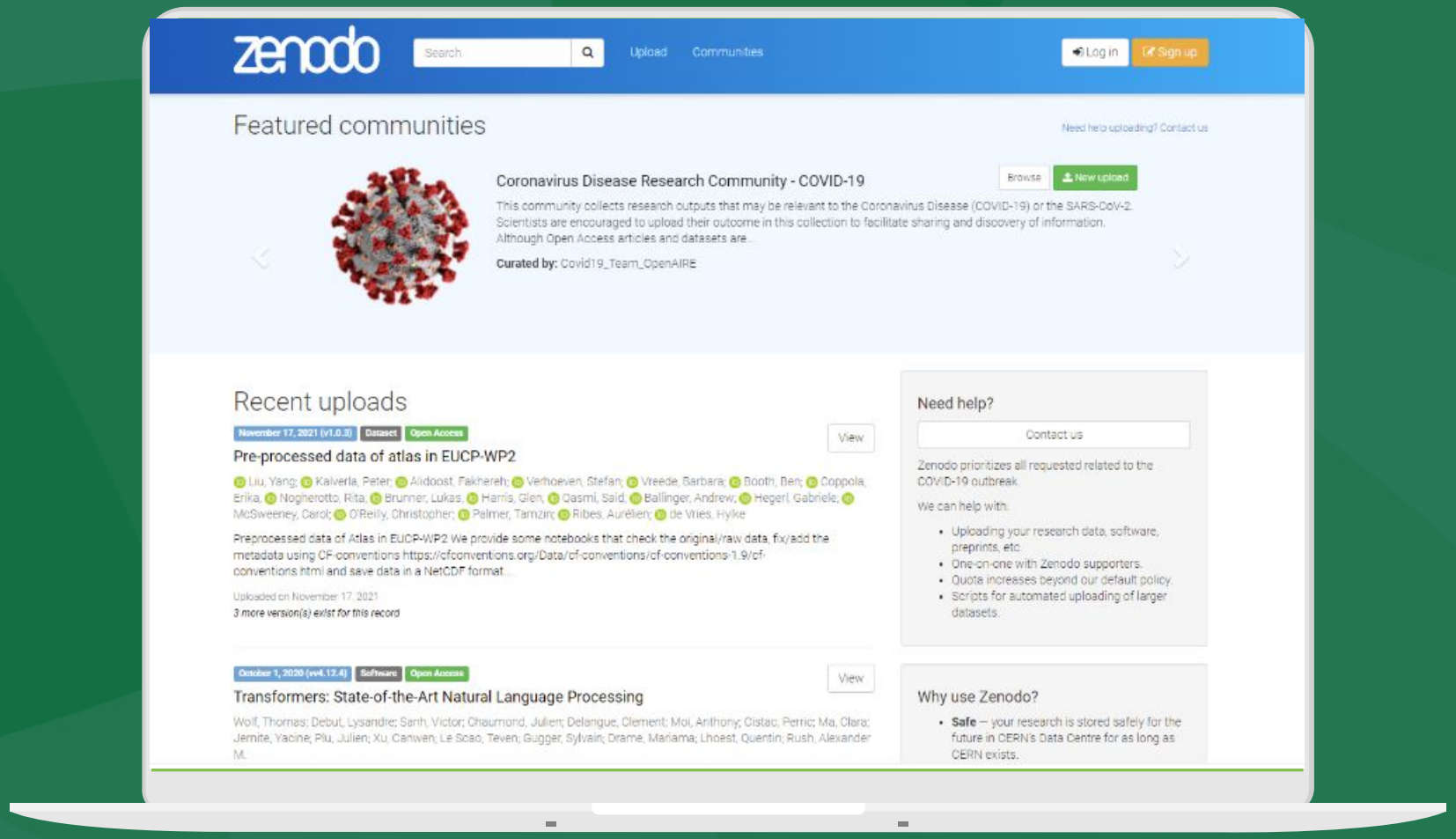
GOLDEN RULES OF GBIF-MEDIATED DATA USE

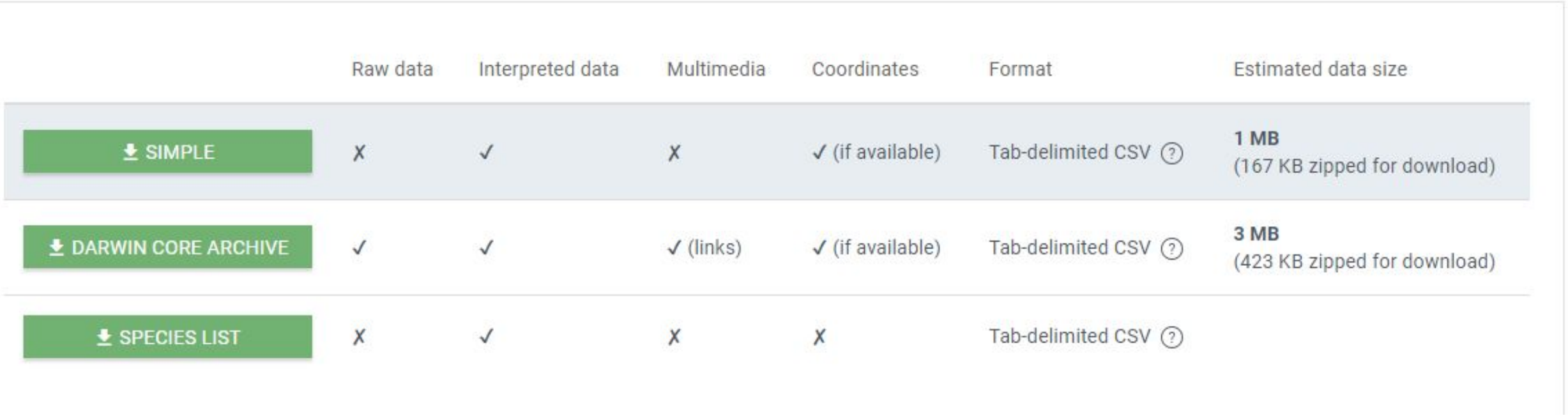
- Must have an account on www.gbif.org
- Must agree to the Data User Agreement - https://www.gbif.org/terms/data-user
- Document how you process your data
- Correctly cite the data you use
  - Guidelines- https://www.gbif.org/citation-guidelines
  - Derived dataset DOIs - https://www.gbif.org/derived-dataset/about



GOLDEN RULES OF GBIF-MEDIATED DATA USE

- Must have an account on www.gbif.org
- Must agree to the Data User Agreement - https://www.gbif.org/terms/data-user
- Document how you process your data
- Correctly cite the data you use
  - Guidelines: https://www.gbif.org/citation-guidelines
  - Derived dataset DOIs
- Deposit used data in a public repository e.g. Zenodo





DATA DOWNLOADS

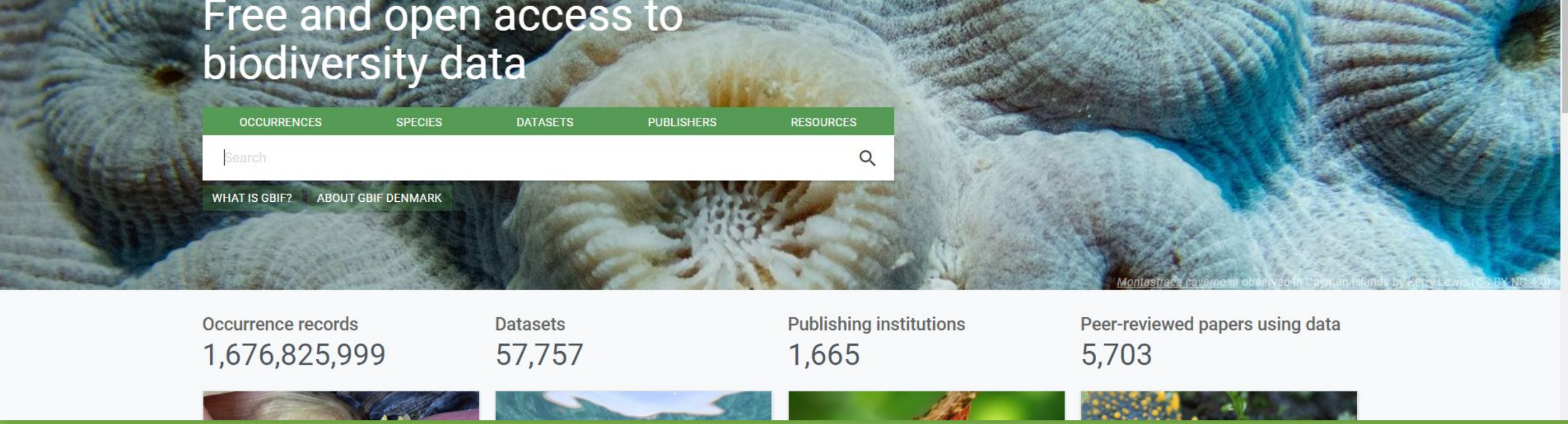
Data can be downloaded in three formats

Simple: Tab delimited CSV. Only contains the data after GBIF interpretation. No multimedia included. More information about CSV

Darwin Core Archive: The Darwin Core Archive (DwC-A) contains both the original data as publisher provided it and the GBIF interpretation. Links (but not files) to multimedia included. More information about DwC-A

Species list: Tab delimited CSV with the distinct list of names in the search result.

GBIF



WEBSITE OVERVIEW

www.gbif.org

GBIF

Exercise 1: Navigating www.gbif.org

What are the total number of occurrences for Tongatapu islands in Tonga?

How many records for the kingdom Plantae are there on the islands?

How many of these records are from the BID programme?

How many of these records are under a CC-BY licence?

How many have images?

Exercise 1: Navigating www.gbif.org

What are the total number of occurrences for Tongatapu islands in Tonga? (**3,372** - 7th Dec 2021 only GADM, **3,362** - 7th Dec 2021 with GADM and country filter)

How many records for the kingdom Plantae are there on the islands? (459 - 7th Dec2021)

How many of these records are from the BID programme? (36 - 7th Dec2021)

How many of these records are under a CC-BY licence? (0 - 7th Dec2021)

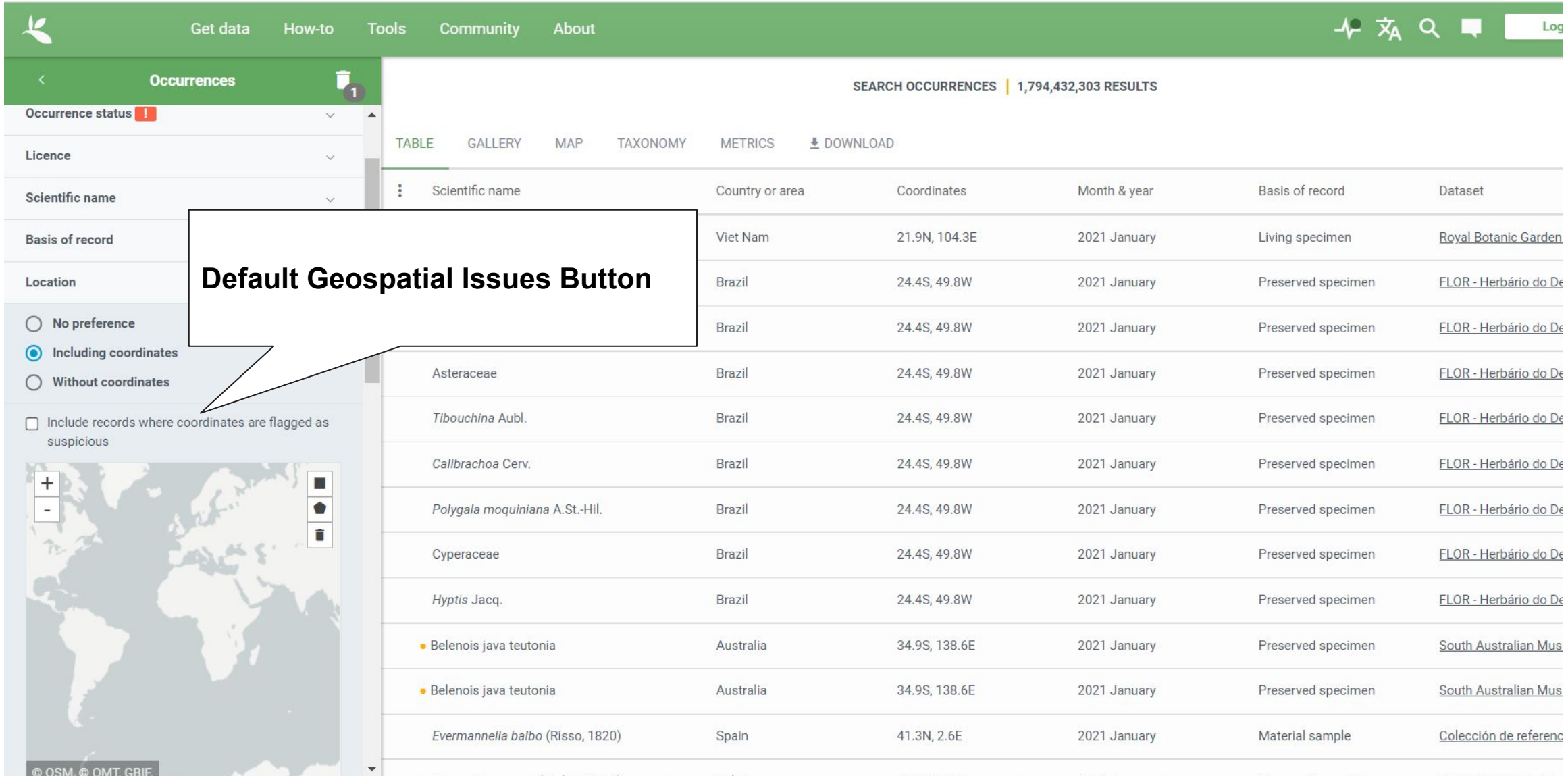
How many have images? (0 - 7th Dec2021)

Common Data Quality Issues

John Waller | Data Analyst

GBIF | Global Biodiversity Information Facility

Your **GBIF download** will not always be ‘perfect’ for what you want do with it. **There are a few things you should be aware of…**

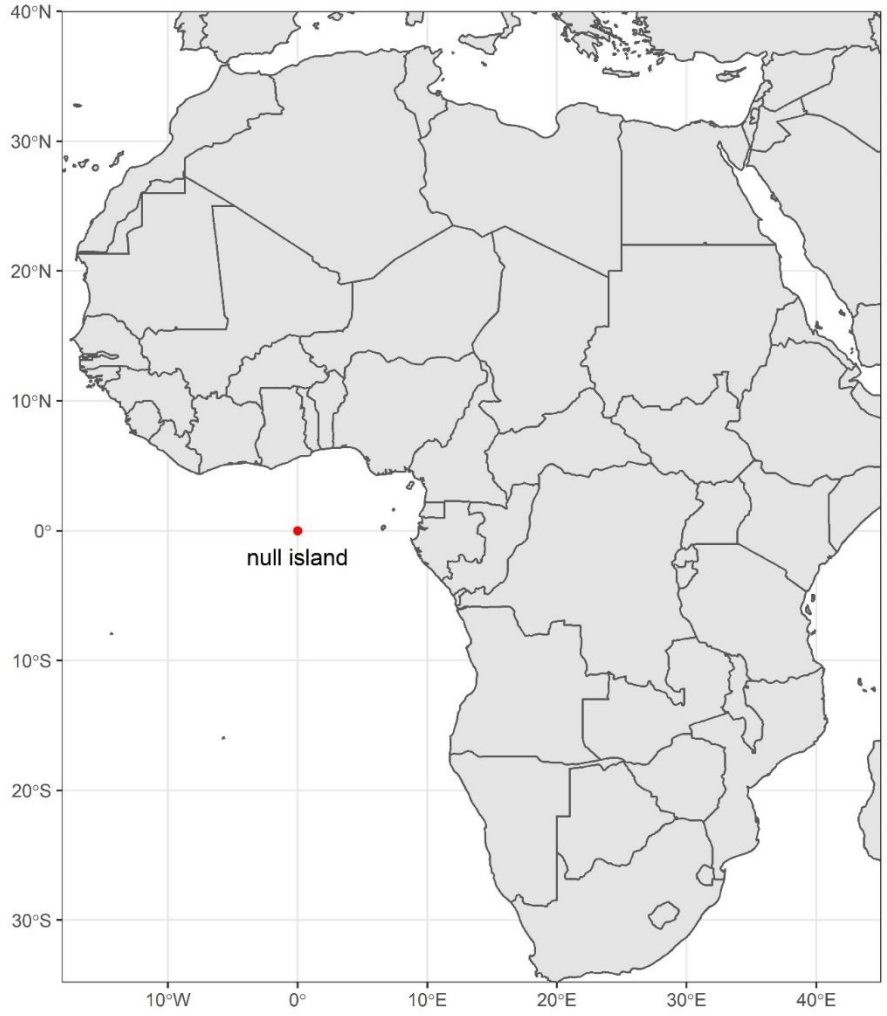

Get data
How-to
Tools
Community
About
Occurrences
Occurrence status
Licence
Scientific name
Basis of record
Location
No preference
Including coordinates
Without coordinates
Include records where coordinates are flagged as suspicious
Default Geospatial Issues Button
SEARCH OCCURRENCES | 1,794,432,303 RESULTS
TABLE
GALLERY
MAP
TAXONOMY
METRICS
DOWNLOAD
Scientific name
Country or area
Coordinates
Month & year
Basis of record
Dataset
Viet Nam 21.9N, 104.3E 2021 January Living specimen Royal Botanic Garden
Brazil 24.4S, 49.8W 2021 January Preserved specimen FLOR - Herbário do De
Brazil 24.4S, 49.8W 2021 January Preserved specimen FLOR - Herbário do De
Asteraceae Brazil 24.4S, 49.8W 2021 January Preserved specimen FLOR - Herbário do De
Tibouchina Aubl. Brazil 24.4S, 49.8W 2021 January Preserved specimen FLOR - Herbário do De
Calibrachoa Cerv. Brazil 24.4S, 49.8W 2021 January Preserved specimen FLOR - Herbário do De
Polygala moquiniana A.St.-Hil. Brazil 24.4S, 49.8W 2021 January Preserved specimen FLOR - Herbário do De
Cyperaceae Brazil 24.4S, 49.8W 2021 January Preserved specimen FLOR - Herbário do De
Hyptis Jacq. Brazil 24.4S, 49.8W 2021 January Preserved specimen FLOR - Herbário do De
Belenois java teutonia Australia 34.9S, 138.6E 2021 January Preserved specimen South Australian Mus
Belenois java teutonia Australia 34.9S, 138.6E 2021 January Preserved specimen South Australian Mus
Evermannella balbo (Risso, 1820) Spain 41.3N, 2.6E 2021 January Material sample Colección de referenc

Default geospatial issues

GBIF removes common geospatial issues by default if you choose to have data with a location.

- **Zero coordinate** : Coordinates are exactly (0,0). null island
- **Country coordinate mismatch** : The coordinates fall outside of the given country's polygon.
- **Coordinate invalid** : GBIF is unable to interpret the coordinates.
- **Coordinate out of range** : The coordinates are outside of the range for decimal lat/lon values ((-90,90), (-180,180)).

GBIF removes zero coordinates (0,0) “null island”



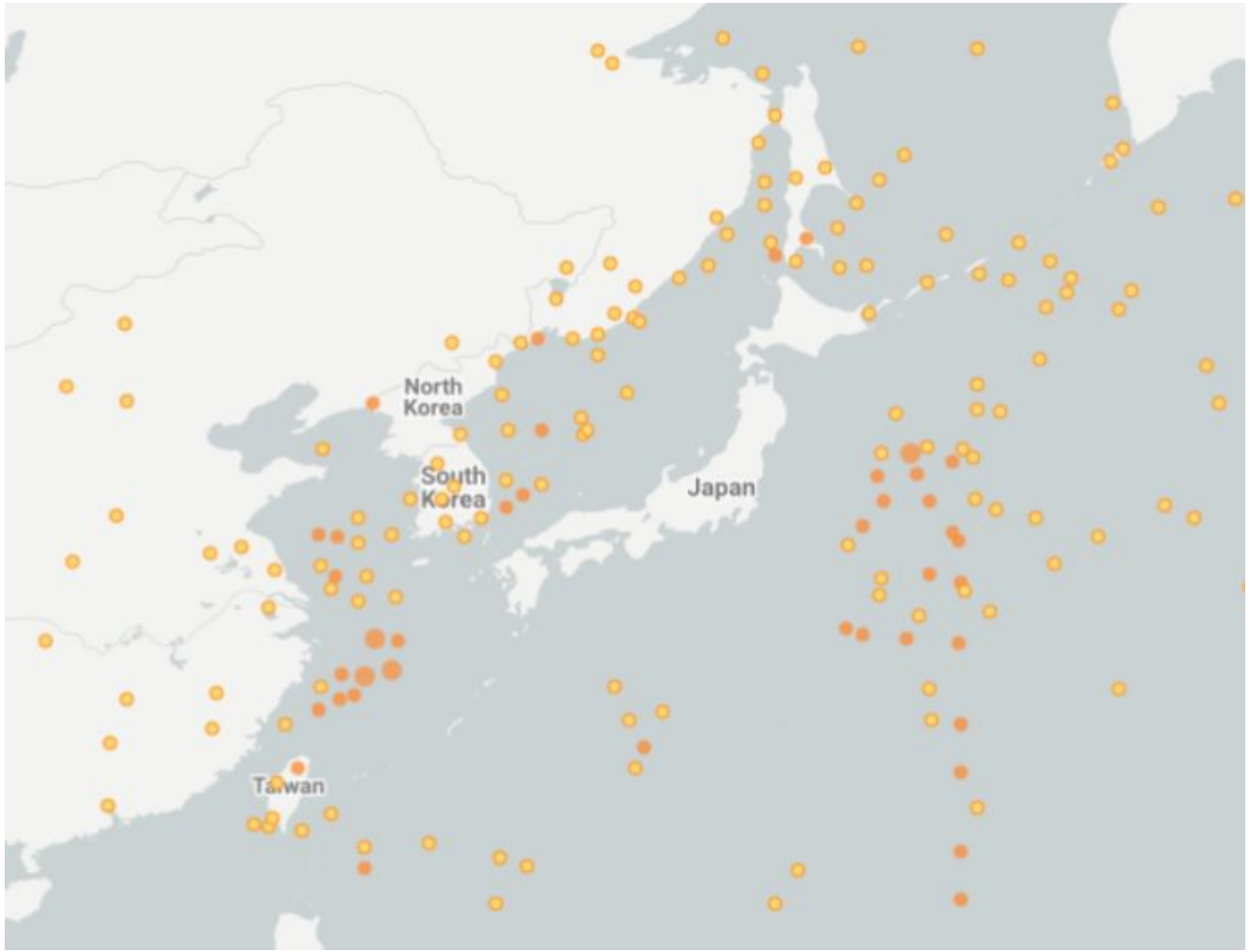
https://www.gbif.org/occurrence/map?issue=ZERO_COORDINATE

GBIF

GBIF removes country coordinate mismatch

GBIF removes records that **do not match their countrycode**.

All of **these records** claim to be located in Japan.



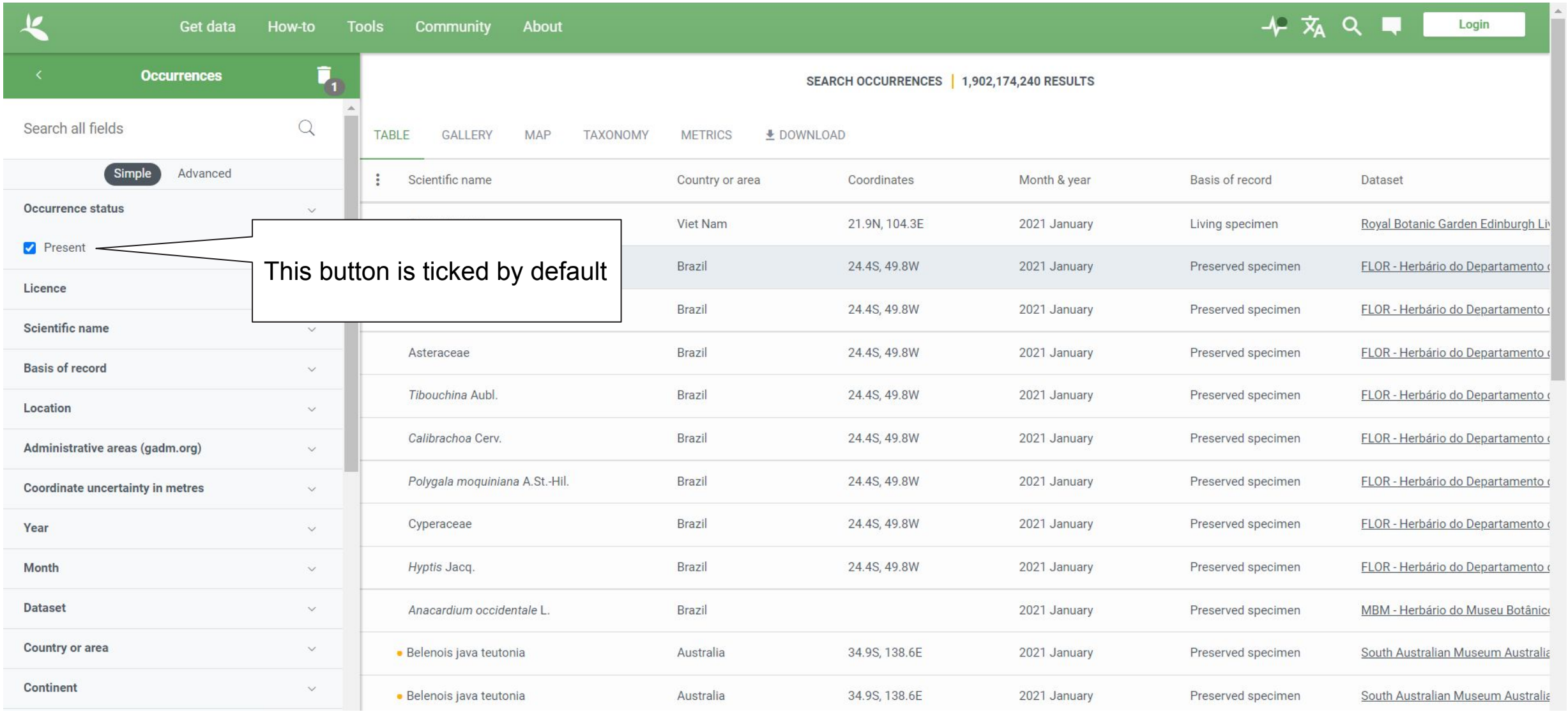
https://www.gbif.org/occurrence/search?issue=COUNTRY_COORDINATE_MISMATCH

GBIF removes absence records

Sometimes data publishers will include **absence records** (where they verify that a species is not present). Most of users don't want these records.

```
gbif_download %>%
filter(occurrenceStatus == "PRESENT")
```

https://www.gbif.org/occurrence/search?occurrence_status=present


Get data
How-to
Tools
Community
About
Login
Occurrences
1
Search all fields
Simple
Advanced
Occurrence status
Present
This button is ticked by default
Licence
Scientific name
Basis of record
Location
Administrative areas (gadm.org)
Coordinate uncertainty in metres
Year
Month
Dataset
Country or area
Continent
SEARCH OCCURRENCES | 1,902,174,240 RESULTS
TABLE
GALLERY
MAP
TAXONOMY
METRICS
DOWNLOAD
Scientific name
Country or area
Coordinates
Month & year
Basis of record
Dataset
Viet Nam
21.9N, 104.3E
2021 January
Living specimen
Royal Botanic Garden Edinburgh Li
Brazil
24.4S, 49.8W
2021 January
Preserved specimen
FLOR - Herbário do Departamento
Brazil
24.4S, 49.8W
2021 January
Preserved specimen
FLOR - Herbário do Departamento
Asteraceae
Brazil
24.4S, 49.8W
2021 January
Preserved specimen
FLOR - Herbário do Departamento
Tibouchina Aubl.
Brazil
24.4S, 49.8W
2021 January
Preserved specimen
FLOR - Herbário do Departamento
Calibrachoa Cerv.
Brazil
24.4S, 49.8W
2021 January
Preserved specimen
FLOR - Herbário do Departamento
Polygala moquiniana A.St.-Hil.
Brazil
24.4S, 49.8W
2021 January
Preserved specimen
FLOR - Herbário do Departamento
Cyperaceae
Brazil
24.4S, 49.8W
2021 January
Preserved specimen
FLOR - Herbário do Departamento
Hyptis Jacq.
Brazil
24.4S, 49.8W
2021 January
Preserved specimen
FLOR - Herbário do Departamento
Anacardium occidentale L.
Brazil
2021 January
Preserved specimen
MBM - Herbário do Museu Botânic
Belenois java teutonia
Australia
34.9S, 138.6E
2021 January
Preserved specimen
South Australian Museum Australia
Belenois java teutonia
Australia
34.9S, 138.6E
2021 January
Preserved specimen
South Australian Museum Australia

Other issues **you have to filter yourself**...

Fossils and Living Specimens

GBIF has **Fossils** and **Living Specimens** (usually a plant inside a botanical garden or sometimes and animal in a zoo).

```
gbif_download %>%
filter(!basisOfRecord  %in%
c("FOSSIL_SPECIMEN","LIVING_SPECIMEN"))
```

establishmentMeans

dwc:establishmentMeans : The process by which the biological individual(s) represented in the Occurrence became established at the location.

```
gbif_download %>%
filter(!establishmentMeans %in% c("MANAGED",
"INTRODUCED", "INVASIVE", "NATURALISED"))
```

Unfortunately not used very often.

https://terms.tdwg.org/wiki/dwc:establishmentMeans

GBIF

Old Records

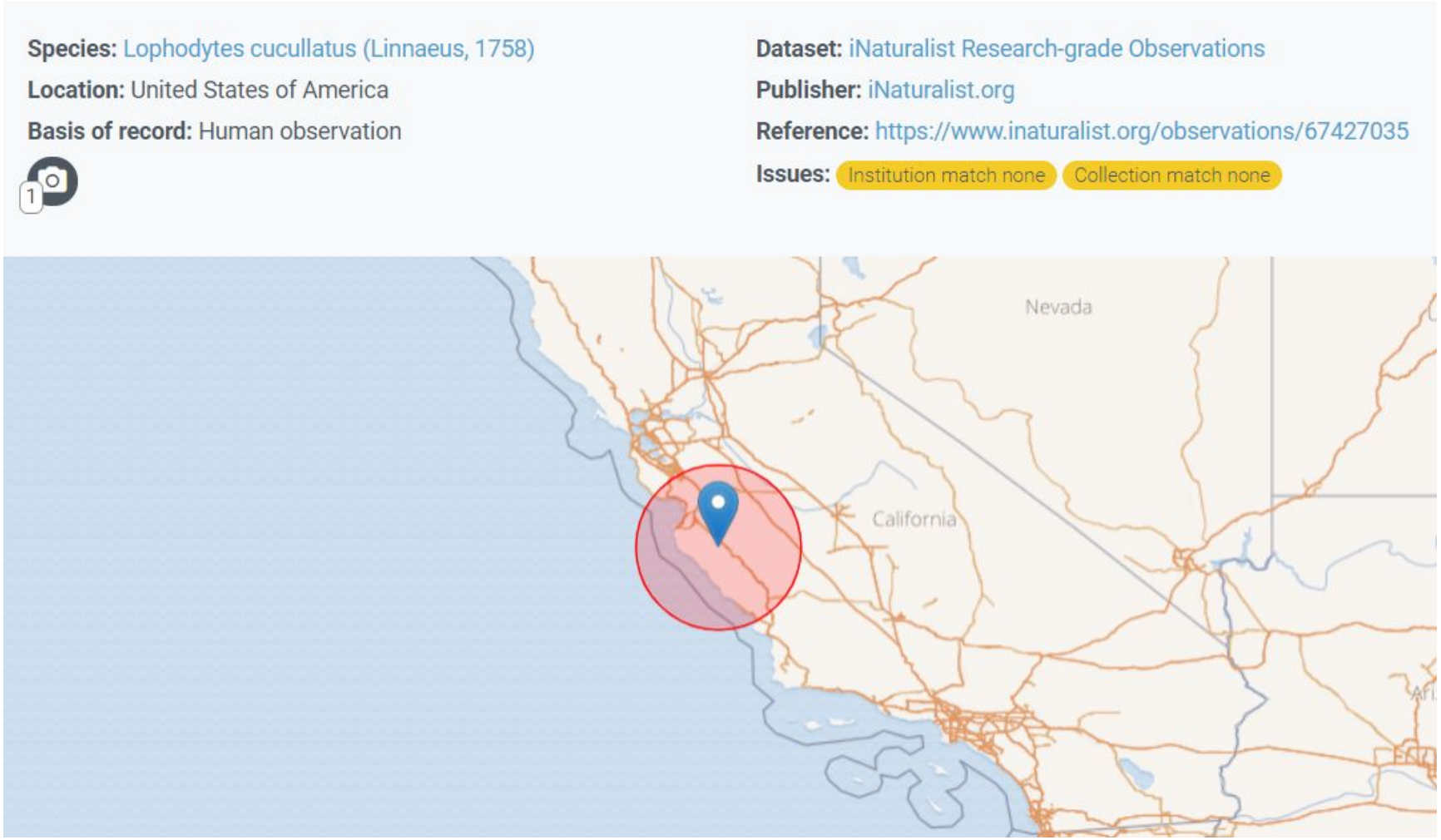
GBIF has many museum records that might be **older than what is desired** for some studies.

```
gbif_download %>%
filter(year >= 1900)
```

https://www.gbif.org/occurrence/search?year=1000,1650&occurrence_status=present

GBIF

Uncertain location example



https://www.gbif.org/occurrence/3017942707

GBIF

Uncertain location

Often you will want to be sure that the coordinates give a certain location and are not really 1000s of km away from where the organism was observed or collected.

```
gbif_download %>%
filter(coordinatePrecision > 0.01 |
is.na(coordinatePrecision)) %>%
filter(coordinateUncertaintyInMeters < 10000 |
is.na(coordinateUncertaintyInMeters))
```

I recommend not filtering out missing values, since the value is often not filled in by publishers if they think the occurrence is fairly certain (from a GPS).

Bad default values for coordinate uncertainty

```
gbif_download %>%
filter(!coordinateUncertaintyInMeters %in%
c(301,3036,999,9999))
```

There are a few “fake” values for coordinate uncertainty that you should be aware of. These values are errors produced by geocoding software and do not represent real uncertainty values. In the case of **301**, the uncertainty is often much-much greater than 301 and actually represents a **country centroid**.

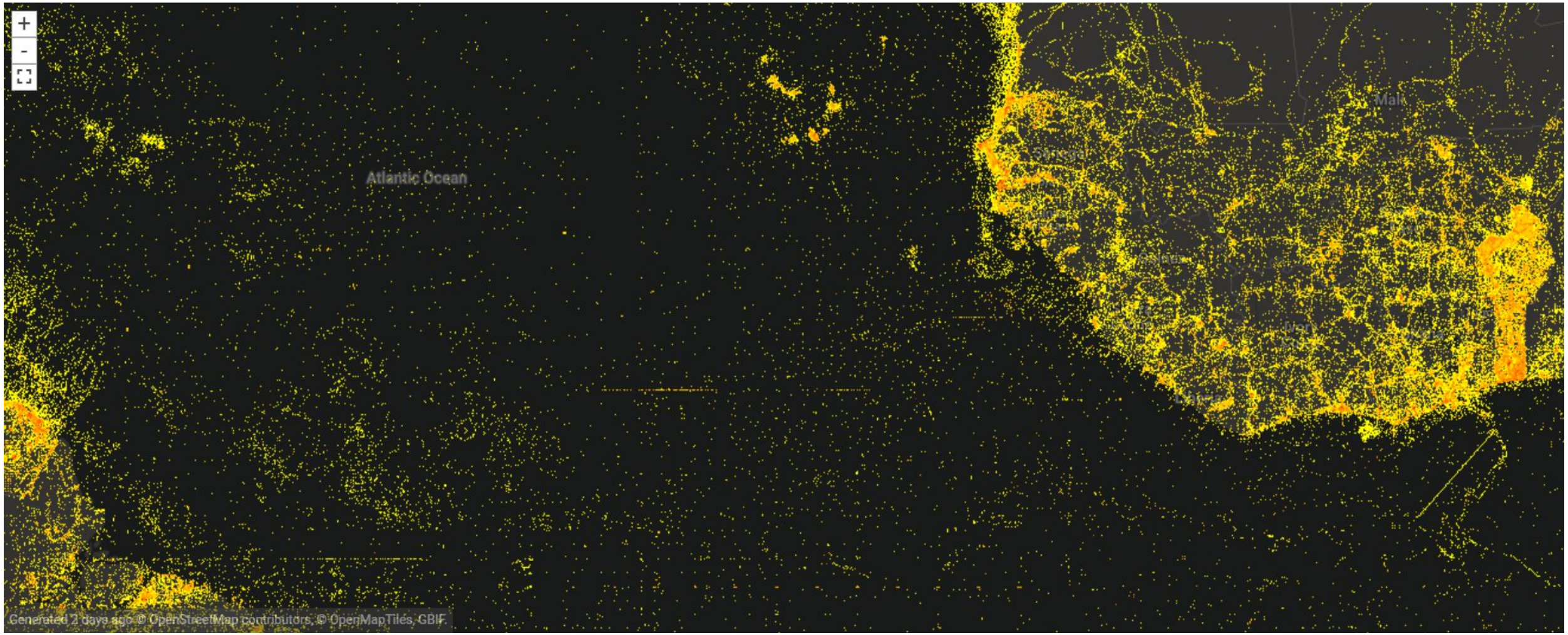
Points along the equator or prime meridian

Some publishers consider zero and NULL to be equivalent, empty latitude and longitude end up being plotted along these two lines.

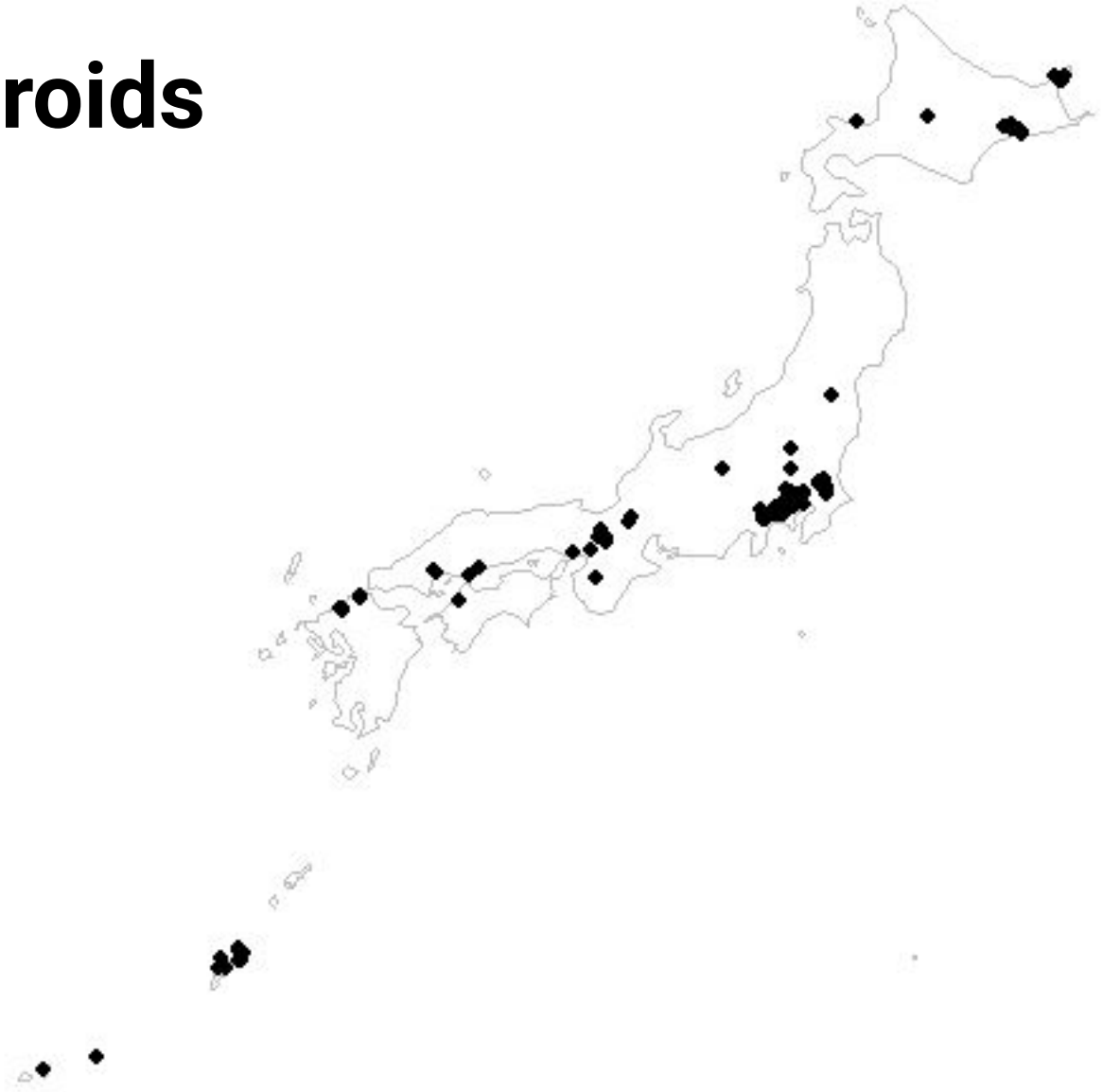
```
gbif_download %>%
filter(!decimalLatitude == 0 |
!decimalLongitude == 0)
```

https://www.gbif.org/ (look at main map for patterns)

GBIF


Atlantic Ocean
Mali
Generated 2 days ago © OpenStreetMap contributors, © OpenMapTiles, GBIF

Country Centroids



Retrospective geo-coding

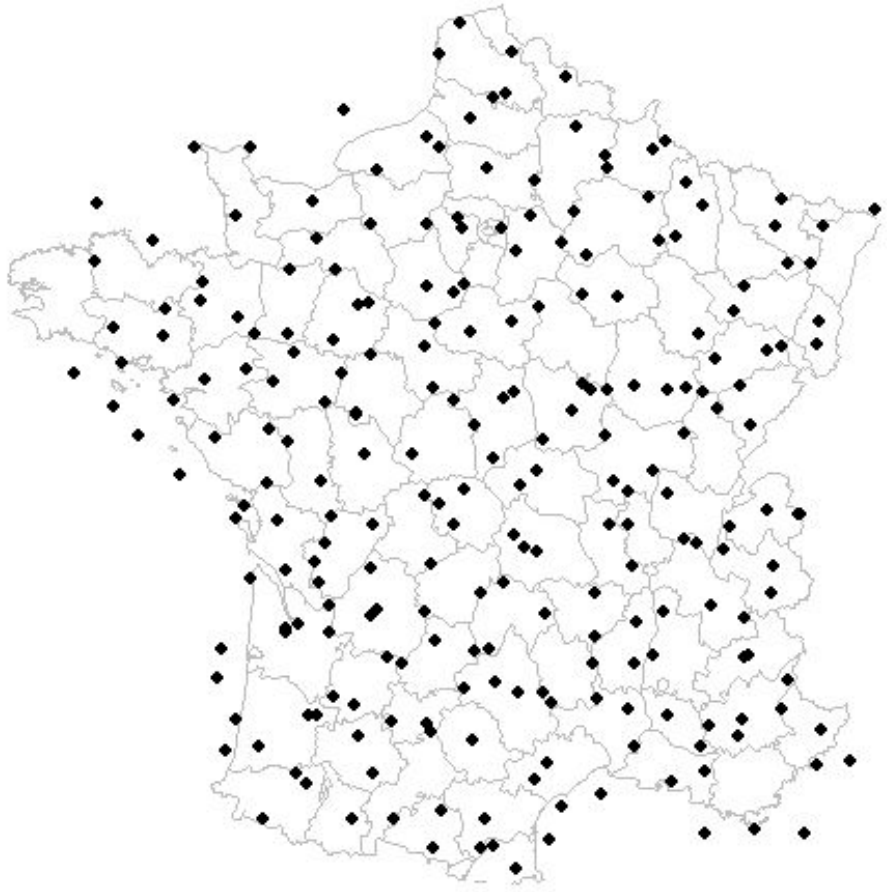
Retrospective geocoding is the process where lat-lon values are given to older records which only have **locality information**.

Locality information is sometimes only a country, city or a text description like

“*10 miles SW of main road, Austin TX*”.

Often museum records (preserved specimens), will have been retrospectively geocoded.

Gridded datasets



https://www.gbif.org/dataset/c779b049-28f3-4daf-bbf4-0a40830819b6

GBIF

Gridded datasets filtering

Most publishers of gridded datasets actually fill in one of the following columns:

- coordinateuncertaintyinmeters
- coordinateprecision
- footprintwkt (only in dwca downloads)

So filtering by these columns **can be a good way to remove gridded datasets**.

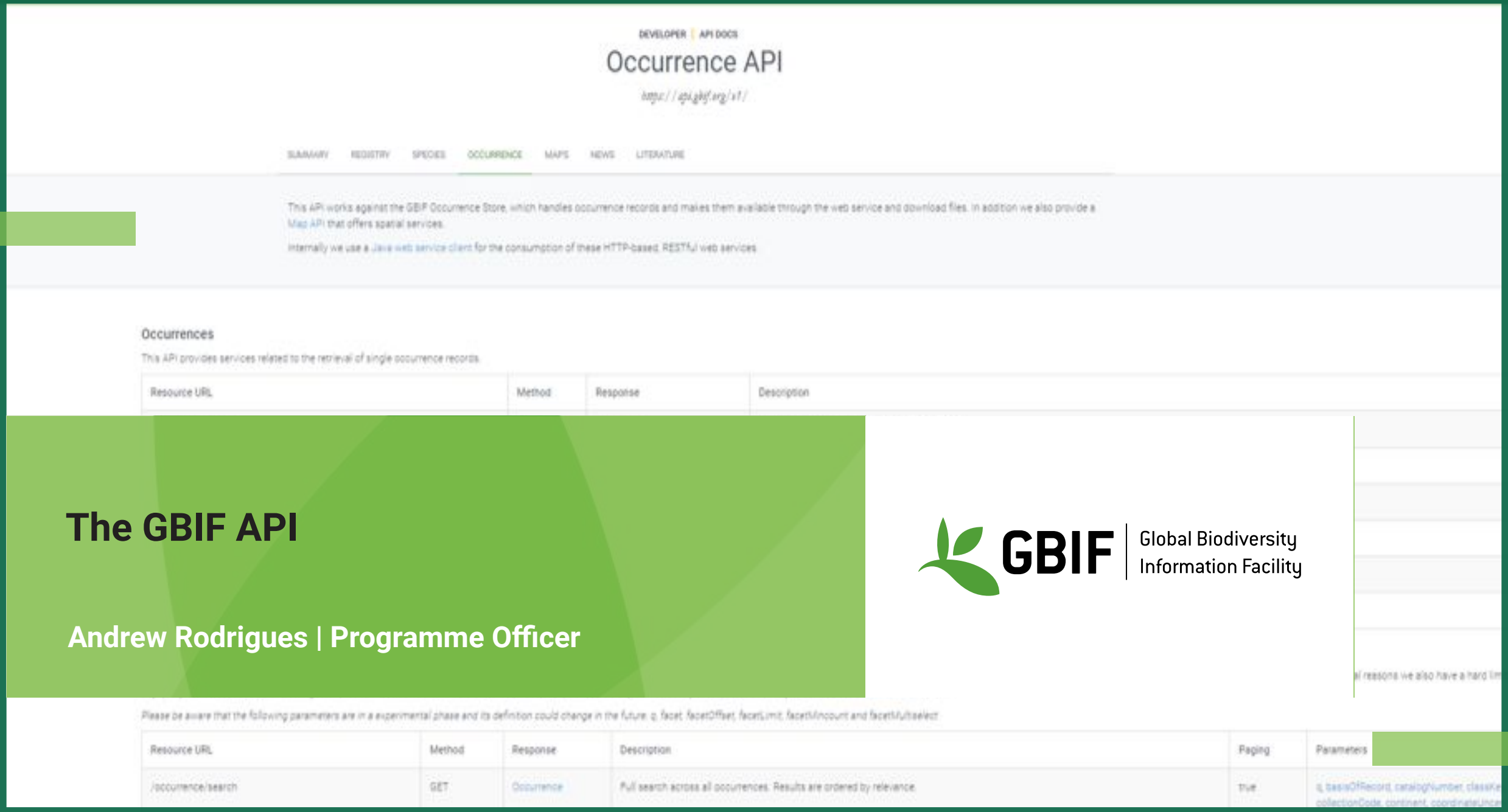
GBIF has an experimental API for identifying datasets which exhibit a certain about of "griddyness". You can read more here.

Exercise 2: Filtering data for improved data quality

Using www.gbif.org filter the data for *Calopteryx splendens* in the following way:

- *Filter for records with coordinate uncertainty between 0 and 10,000m*
- *Filter for records between 1955 and 2017*
- *Exclude occurrence records where the establishment mean is indicated as managed, introduced or invasive*

How many records did you have at the start?
How many do you have after filtering?
How did you find the taxon?
What are the limitations of the filters?



The GBIF API

Andrew Rodrigues | Programme Officer

GBIF | Global Biodiversity Information Facility

What is the GBIF API?

GBIF Application Programming Interface (API) gives users access to GBIF databases in a **secure way**

Usually the main reason you would want to use an API is because you want **software**, to interact with GBIF somehow

The GBIF API can be accessed via:

- A web browser by visiting a URL e.g. https://api.gbif.org/v1/species/match?name=Passer domesticus
- Or using a command line program called curl that you need to install

GBIF has a few API group/namespace:

- **Registry API** - makes all registered Datasets, Installations, Organizations, Nodes, and Networks discoverable.
- **Species API** - works against data kept in the GBIF Checklist Bank which taxonomically indexes all registered checklist datasets in the GBIF network.
- **Occurrence API** - works against the GBIF Occurrence Store, which handles occurrence records and makes them available through the web service and download files.
- **Maps API** - web map tile service making it straightforward to visualize GBIF content on interactive maps, and overlay content from other sources.
- **Literature API** - search for literature indexed by GBIF, including peer-reviewed papers, citing GBIF datasets and downloads.

What is the GBIF API?

The **basic pattern** of an API call:

- **base url** : this will always be **https://api.gbif.org/v1/**
- **api** : this is the GBIF API group/namespace you want to query.
- **function** : the functionality you want to use.
- **parameter** : the parameters for your API call. A **?** is sometimes used.
- **query** : the query you fill in. Sometimes will be free text and sometimes will be a predefined argument.

Example

https://api.gbif.org/v1/species/match?name=Passer domesticus

Exercise 3 - Finding GBIF taxonkeys

- Taxon keys are issued at the species, genus family, order, phylum and kingdom level.
- Unique identifiers are issued to accepted names with synonyms of those accepted names issued the same identifier. Usage keys vs acceptedusagekeys
- Allow for user to ensure that they are collecting all the data they need
- Also facilitate multiple species downloads
- Taxon keys can be found through:
  - www.gbif.org
  - Species API
  - Species Matching tool - https://www.gbif.org/tools/species-lookup

https://api.gbif.org/v1/species/match?name=Calopteryx%20splendens

Exercise 3 - Finding GBIF taxonkeys

Using the Species API find the GBIF **taxonkeys** for these scientific names:

- *Lepus saxatilis F.Cuvier, 1823*
- Aves
- Magnoliophyta
- *Aegithalos caudatus* (Linnaeus, 1758)

What are the taxonomic status of each?

https://api.gbif.org/v1/species/match?name=Calopteryx%20splendens

Exercise 3 - Finding GBIF taxonkeys

Using the Species API find the GBIF **taxonkeys** for these scientific names:

- *Lepus saxatilis F.Cuvier, 1823* = 2436775 (ACCEPTED)
-
- Aves = 212 (ACCEPTED) Note: that this is not a species, so if you want occurrences for an entire group you just need one taxonkey and not a list of every species in the group.
-
- Magnoliophyta = 49 (SYNONYM). Note: that if you used the API directly GBIF will give you the acceptedUsageKey : 7707728, which is Tracheophyta. This is the accepted name of this group. If you decided to just use the old name, you would be missing millions of occurrences so be careful.
-
- *Aegithalos caudatus* (Linnaeus, 1758) = 2495000 (DOUBTFUL) Note: this is an interesting case because this "doubtful" name has millions of occurrences tied to it. There is some apparently interesting taxonomic history behind this case…

What are the taxonomic status of each?

https://api.gbif.org/v1/species/match?name=Calopteryx%20splendens

R and rgbif

John Waller | Data Analyst & rgbif maintainer

GBIF | Global Biodiversity Information Facility

is a **programming language.**

It is commonly used for **statistics** and **research**.

There are **thousands** of R packages.

GBIF

```r
# basic math (use # for comments)
x <- 2 # assign a variable
x + 2
x*x
(x - 10)/2

# some data types
v <- c(1,2,3,4)
l <- list(1,"cat",c(1,2,3))
d <- data.frame(pets = c("dog","cat"),num = c(1,2))
pet <- "dog"
class(v) # use class to see type
```

GBIF

```r
# functions
print("dog")
class(1)
getwd()
?getwd # get help

# write your own function
test_fun <- function(a,b) a + b
test_fun(2,2)
```

GBIF

```r
# R packages are collections of functions
install.packages("tidyverse")
install.packages("rgbif")
install.packages("CoordinateCleaner",dependencies = TRUE)
# load packages
library(tidyverse)
library(rgbif)
library(CoordinateCleaner)

.libPaths() # where the packages were installed
rgbif:: # type this in Rstudio to get list of functions
```

GBIF

```r
d <- data.frame(x=c(1,2,3))
View(d) # view like in excel


library(dplyr) # for %>% and filter
"dog" %>% print() # pipe
print("dog") # same as above


# useful for filters
d %>%
filter(x > 1) %>%
glimpse()
```

GBIF

```r
# read in an external table
library(readr)
table <- read_tsv("C:/Users/John/Desktop/some_file.tsv")

# basic data manipulation
library(dplyr)
d <- data.frame(x=c(1,2,3),y=c("cat","dog","dog"))
d$x # select single column
d %>% pull(x) # select single column

d %>%
group_by(y) %>%
count()
```

GBIF


RStudio®

rgbif is a **R package.**

rgbif uses the **GBIF API** to access GBIF mediated data from within R.

It is useful for to **downloading** and **looking up species names** among other things.

https://docs.ropensci.org/rgbif/

GBIF

```
library(rgbif)
name_backbone("Lepus saxatilis") # look up a taxonkey
occ_search(taxonKey=2436775) # preview some records


# preview a download request
occ_download_prep(pred("taxonKey"),2436775))
# run an actual download
k <-occ_download(pred("taxonKey"),2436775))
occ_download_wait(k) # wait for a download to finish
# download list of species
occ_download(pred_in(("taxonKey"),c(2436775,10903982))
```

https://docs.ropensci.org/rgbif/

GBIF

Exercise 4: Setting up R Environment

```
# only need to run once
install.packages("tidyverse")
install.packages("rgbif")
install.packages("CoordinateCleaner")
# run every time you restart Rstudio
library(tidyverse)
library(rgbif)
library(CoordinateCleaner)
```

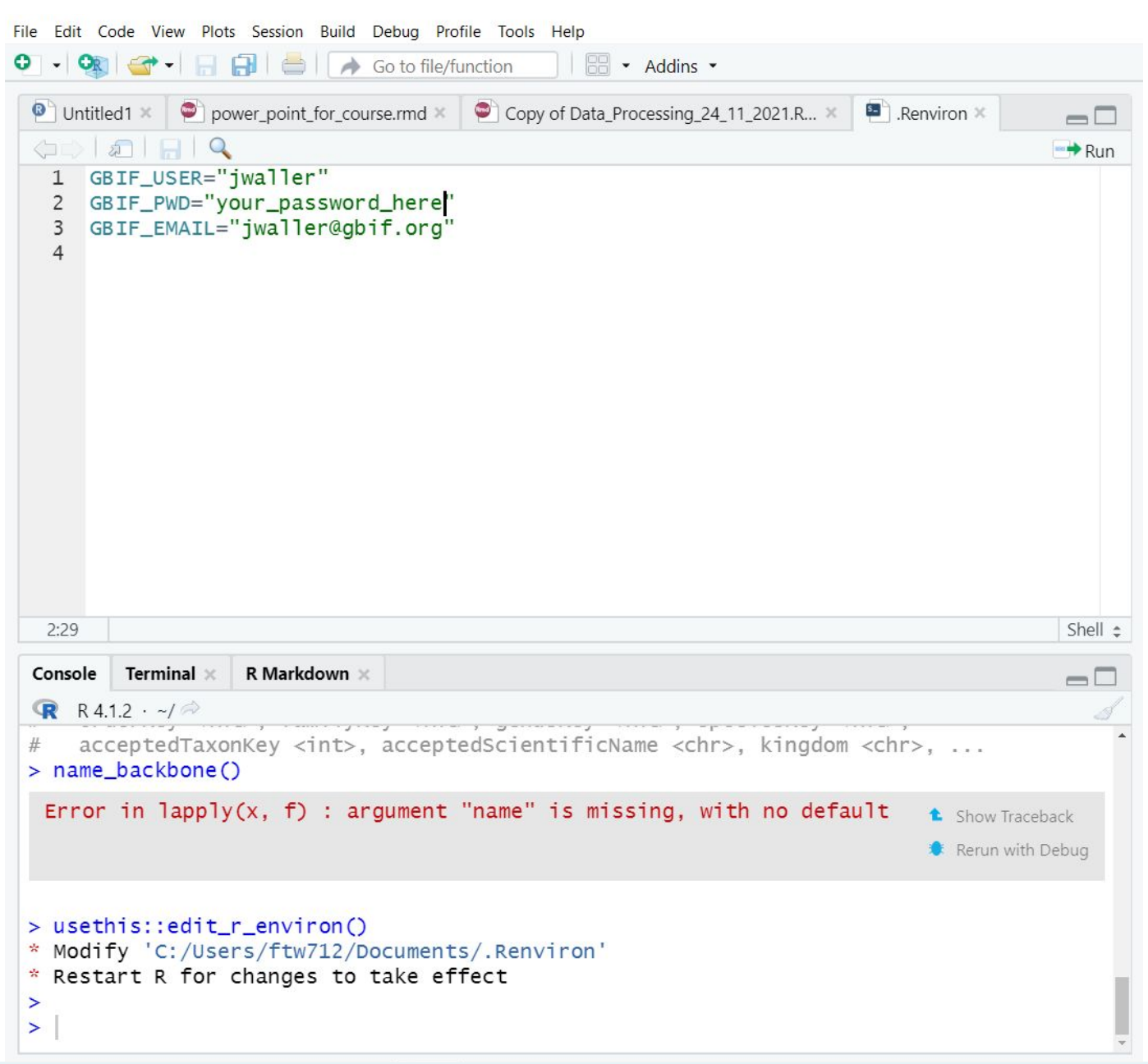
Using **RStudio**, run the set up code above.

GBIF

Setting up R Environment (optional)

```
install.packages("usethis")
usethis::edit_r_environ()
# Edit your .Renviron to look like this but with your information:


GBIF_USER="jwaller"
GBIF_PWD="safe_fake_password123"
GBIF_EMAIL="jwaller@gbif.org"
```


File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Untitled1
power_point_for_course.rmd
Copy of Data_Processing_24_11_2021.R...
.Renviron
Run
1 GBIF_USER="jwaller"
2 GBIF_PWD="your_password_here"
3 GBIF_EMAIL="jwaller@gbif.org"
4
2:29
Shell
Console
Terminal
R Markdown
R 4.1.2 · ~/
acceptedTaxonKey <int>, acceptedScientificName <chr>, kingdom <chr>, ...
> name_backbone()
Error in lapply(x, f) : argument "name" is missing, with no default
Show Traceback
Rerun with Debug
> usethis::edit_r_environ()
* Modify 'C:/Users/ftw712/Documents/.Renviron'
* Restart R for changes to take effect
>
>

Exercise 5: Downloading a dataset using rgbif

Download a dataset (simple csv) using **rgbif** that has the following properties:

1. Taxon is *Lepus saxatilis*
2. Found in South Africa (ZA)
3. That is a preserved specimen or human observation
4. That has latitude and longitude coordinates
5. Does not have common geospatial issues

In other words:
Lepus saxatilis + in ZA + specimen or human observation + has coordinates + no geoissues

```
name_backbone("Lepus saxatilis") # look up a taxonkey
```

GBIF

Exercise 5: Downloading a dataset using rgbif

```
library(rgbif)

user <- ""
pwd <- ""
email <- ""

occ_download(
pred("taxonKey", ?),
pred_in("basisOfRecord",
c('PRESERVED_SPECIMEN','HUMAN_OBSERVATION'),
pred("country", "ZA"),
pred("hasCoordinate", TRUE),
pred("hasGeospatialIssue", FALSE),
format = "SIMPLE_CSV",
user=user,pwd=pwd,email=email
)
```

GBIF

Exercise 5: 1st Step - provide credentials to GBIF

```
user <- ""
pwd <- ""
email <- ""
```

```
install.packages("usethis")

usethis::edit_r_environ()

# Edit your .Renviron to look like this but with your information:

GBIF_USER="jwaller"

GBIF_PWD="safe_fake_password123"

GBIF_EMAIL="jwaller@gbif.org"
```

GBIF

Exercise 5: 2nd Step - Use the occ_download function

```
gbif_download_key <- occ_download(
pred("taxonKey", 2436775),
pred_in("basisOfRecord",c('PRESERVED_SPECIMEN','HUMAN_OBSERVATION'),
pred("country", "ZA"),
pred("hasCoordinate", TRUE),
pred("hasGeospatialIssue", FALSE),
format = "SIMPLE_CSV",
user=user,pwd=pwd,email=email
)
```

Exercise 5: 3rd Step - Importing your download into R

```
# import your download into R
data_download <- occ_download_get(gbif_download_key,
overwrite = TRUE) %>%
occ_download_import()
View(data_download)


# obtain a DOI for your dataset
res <- occ_download_meta(gbif_download_key)
gbif_citation(res)
```

GBIF

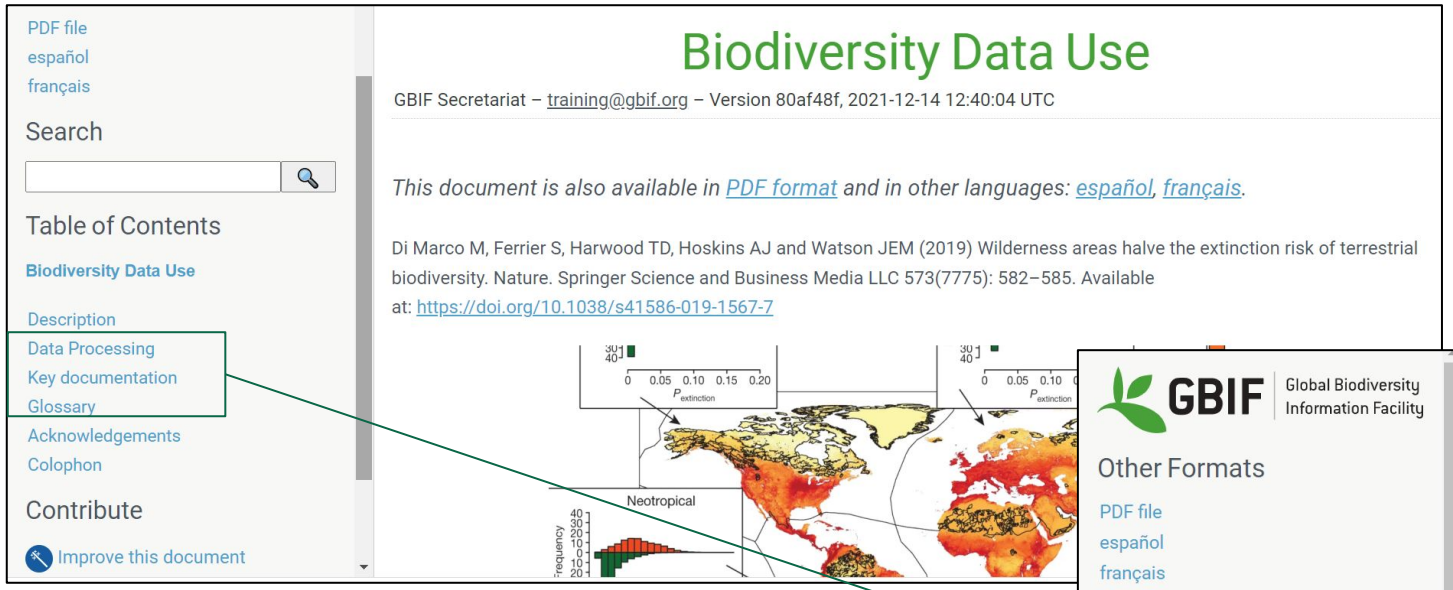
Exercise 6: Downloading a long species list

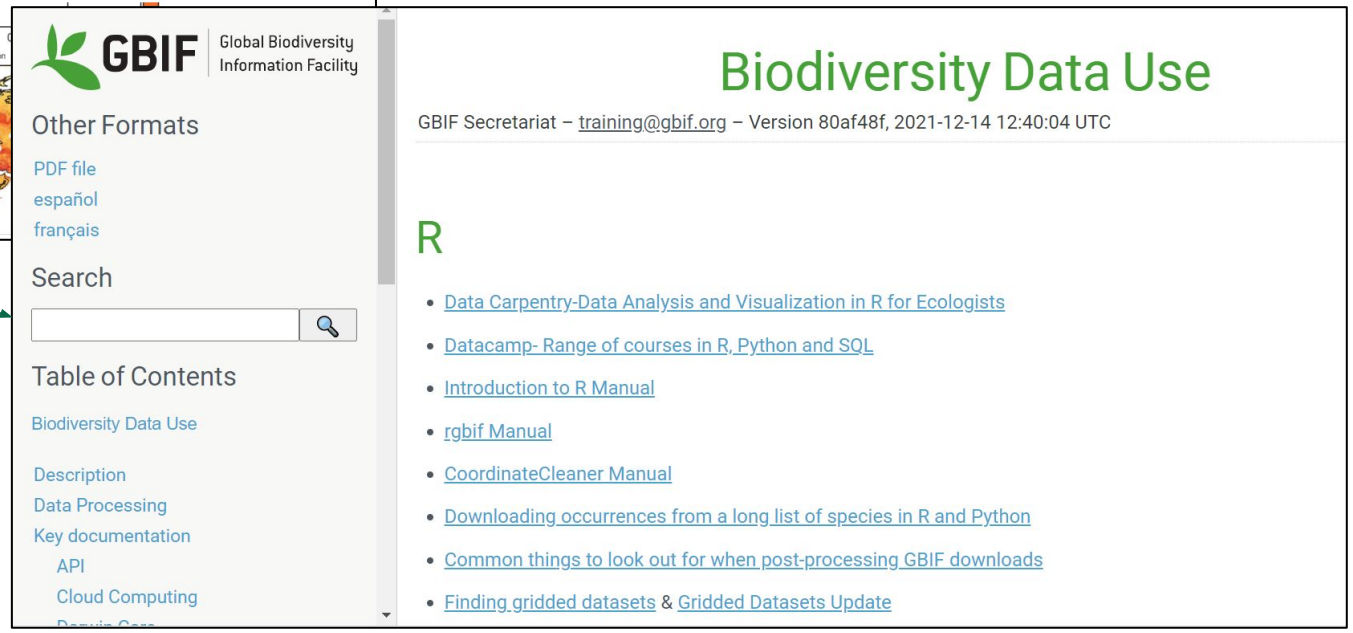
```
gbif_taxon_keys <- c(3189834,3189801,2876099,2888580)

occ_download(
pred_in("taxonKey", gbif_taxon_keys),
pred("hasCoordinate", TRUE),
pred("hasGeospatialIssue", FALSE),
format = "SIMPLE_CSV",
user=user,pwd=pwd,email=email
)
```

GBIF

Resources





Data Processing example
Lemur_catta_project.rmd

Community Forum
https://discourse.gbif.org/g/BID_DataUse

Supplementary Exercise: Data cleaning with R

Clean your *Lepus saxatilis* download by

- *Removing occurrence records where the **establishmentmeans** is indicated as **managed**, **introduced** or **invasive***
- *Filtering **year** for records between 1955 and 2010*
- *Filtering for records with **coordinate uncertainty** of less than 10,000 and **coordinate precision** of greater than 0.01*
- *Removing points within 2 km of **country centroids** and **capital centroids***
- *Removing points within 2 km of a **zoo** or **herbarium***

```r
library(rgbif)
library(dplyr) # for filter and %>%
library(CoordinateCleaner) # for cc_cen,cc_cap,cc_inst

# gbif_download_key <- "0071981-210914110416597"

# first import data
gbif_download <- occ_download_get(gbif_download_key, overwrite = TRUE) %>%
  occ_download_import()

gbif_download %>%
setNames(tolower(names(.))) %>% # set lowercase column names to work with CoordinateCleaner
filter(!establishmentmeans %in% c("MANAGED", "INTRODUCED", "INVASIVE")) %>%
filter(year >= 1955 & year <= 2010) %>%
filter(coordinateprecision < 0.01 | is.na(coordinateprecision)) %>%
filter(coordinateuncertaintyinmeters < 10000 | is.na(coordinateuncertaintyinmeters)) %>%
cc_cen(buffer = 2000) %>% # remove country centroids within 2km
cc_cap(buffer = 2000) %>% # remove capitals centroids within 2km
cc_inst(buffer = 2000) %>% # remove zoo and herbaria within 2km
glimpse() # look at results of pipeline
```

GBIF

Step 1 - Load libraries

This code will load the functions we need to clean the data.

CoordinateCleaner is an R package written specifically for **cleaning GBIF occurrence data**. https://github.com/ropensci/CoordinateCleaner

```
library(rgbif)
library(dplyr) # for filter and %>%
library(CoordinateCleaner) # for cc_cen,cc_cap,cc_inst
```

GBIF

Step 2 - Import Data

This code will import that data from GBIF into R.

Remember that the pipe (**%>%**) just passes the results of one function into another function.

```
# gbif_download_key <- "0071981-210914110416597"
# first import data
gbif_download <- occ_download_get(gbif_download_key, overwrite
= TRUE) %>%
  occ_download_import()
```

Step 3 - Import data and clean column names

Set the column names to lowercase. The dot (“.”) here is special pipe code which refers back to the gbif_download object.

https://magrittr.tidyverse.org/reference/pipe.html

```
gbif_download %>%
setNames(tolower(names(.))) %>% # set lowercase column
names to work with CoordinateCleaner
```

GBIF

Step 4 - filter establishmentMeans

Here we use `filter` from the package **dplyr** to remove not naturally established records. "**!** " means **negation** in R. The operator `%in%` checks if value in the column is in the vector `c("MANAGED", "INTRODUCED", "INVASIVE")`.

```
filter(!establishmentmeans %in% c("MANAGED", "INTRODUCED",
"INVASIVE")) %>%
```

GBIF

Step 5 - filter year

Here we use `filter` to keep only records between 1955 and 2010. The operator `>=` greater than or equal and means `<=` less than or equal. The `&` operator combines and checks that **both conditions** are TRUE.

```
  filter(year >= 1955 & year <= 2010) %>%
```

GBIF

Step 6 - filter uncertain records

Here we use `filter` to keep only certain records. The `|` operator combines and checks that **only one condition** is TRUE. The `is.na` function checks if record is missing. **NA** means missing values in R.

```
filter(coordinateprecision < 0.01 | is.na(coordinateprecision))
%>%
filter(coordinateuncertaintyinmeters < 10000 |
is.na(coordinateuncertaintyinmeters)) %>%
```

Step 7 - filter with CoordinateCleaner

Here we use **CoordinateCleaner** to remove country centroids and records near zoos and botanical gardens.

```
cc_cen(buffer = 2000) %>% # remove country centroids
cc_cap(buffer = 2000) %>% # remove capitals centroids
cc_inst(buffer = 2000) %>% # remove zoo and herbaria
```

GBIF

RESOURCE LINKS

GBIF occurrence search (GBIF occurrence search)
https://data-blog.gbif.org/post/downloading-long-species-lists-on-gbif/ (download)
https://data-blog.gbif.org/post/gbif-filtering-guide/ (filtering guide)
https://data-blog.gbif.org/categories/gbif/ (data blog)
https://data-blog.gbif.org/post/outlier-detection-using-dbscan/ (outlier detection)
https://data-blog.gbif.org/post/gbif-molecular-data-quality/ (metagenomics)