Data Mobilisation from GBIF to the EBV Data Portal for IAS of Union Concern

Notebook 1 - Prototype for creating a occurrence cube of IAS

true

2024-08-20

Introduction

In this notebook we create a cube of occurrence of invasive alien species (IAS) of union concern based on all species records available in the GBIF—the Global Biodiversity Information Facility— until mid August 2024. To do this, we sent a request with the cube specifications (JSON file) via the GBIF API using the occurrence cube software developed by GBIF under the Biodiversity Building Blocks for Policy (B3) project. Details of the GBIF data query are available at DOI 10.15468/dl.gxk3vh.

Note: This series of notebooks is part of the results of Task 3.3 of the Biodiversity Building Blocks for Policy project funded by the European Union's Horizon Europe Research and Innovation Programme (ID No 101059592). Additional notebooks exploring the results and calculating simple metrics are also available in the same repository.

Load Libraries

```
rm(list=ls())
gc()
```

```
##           used (Mb) gc trigger  (Mb) max used  (Mb)
## Ncells 1848748 98.8    5945685 317.6 11612665 620.2
## Vcells 3860433 29.5   54881456 418.8 70740088 539.8
```

```
# load requiered libraries
library(here)
library(httr)
library(httr2)
library(jsonlite)
library(b3gbi)
```

Submit a JSON query

The JSON file containing the query specification has been previously prepared (see the `input/queries` folder). In this case, the cube specification includes the geographical extent (EU27), the spatial resolution

(~1 km), the time period (=>1900), the specieskey of the species of interest (invasive alien species of Union concern (IAS)), and we request the number of occurrences per grid cell, the higher taxonomy and the basis of records. We used the European Environment Agency grid at 10 Km as a reference.

```
# Save GBIF API address
gbif_url <- 'http://api.gbif.org/v1/occurrence/download/request/'

# Set GBIF user name and password. In this example the data is saved in a separate file for security re
gbif_user <- read.csv("C:/gitrepo/credentials.txt", header = FALSE);
my_username <- gbif_user[1,1]; # 'your_username'
my_password <- gbif_user[2,1]; # 'your_password'

# Set the location for the resulting occurrence cube
path_out <- here("output/datacubes/csv/ias/")

# Load the JSON with the query specifications
query_ias <- "query_ias_allyears.json"
```

```
# Request data using GBIF API and your saved query
req <-  request(gbif_url) |>
  req_auth_basic(username = my_username, password = my_password) |>
  req_headers("Content-Type" = "application/json") |>
  req_retry(max_tries = 5) |>
  req_body_file(here(paste0("input/queries/ias/", query_ias)), type = NULL)

# Perform the request and return the response (the response is a download ID)
# response <- req |> req_perform()  |> resp_body_string() # due to the processing time we have pre-subm
response <- "0077925-240506114902167" # Data available from 1900 until 20240815
```

Unpack the IAS occurrence cube and load it into R

```
# Download the cube
download.file(paste0(gbif_url, response, ".zip"), destfile=paste0(path_out, "/", response, ".zip"))

# Or download manually from
paste0(gbif_url, response)
```

```
## [1] "http://api.gbif.org/v1/occurrence/download/request/0077925-240506114902167"
```

```
# Unzip
unzip(paste0(path_out, "/", response,".zip"), exdir=paste0(path_out,"/"))
```

The resulting occurrence cube is in CSV format. The number of species occurrence is aggregated per cell based on the (coded) reference grid. There are several ways to load the resulting cube. Two examples are shown here. One is using the CSV read command and the other is using the `b3gbi` library.

```
# Load GBIF data cube using a simple CSV read function
c1 <- read.csv(paste0(path_out, "/", response,".csv"), sep = "\t")
colnames(c1)
```

```
##  [1] "yearmonth"     "eeacellcode"   "basisofrecord" "datasetkey"
##  [5] "countrycode"   "specieskey"    "species"       "family"
##  [9] "class"         "occurrences"
```

Alternatively, we can use the `process_cube` function from the `b3gbi` library.

```
# Prepare cube using b3gbi
c2 <- process_cube(paste0(path_out, "/", response,".csv"))
print(c2)
```

```
##
## Processed data cube for calculating biodiversity indicators
##
## Date Range: 1900 - 2023
## Single-resolution cube with cell size 10km ^2
## Number of cells: 36071
## Coordinate range:
## xmin xmax ymin ymax
##  262  710  143  557
##
## Total number of observations: 3502320
## Number of species represented: 77
## Number of families represented: 50
##
## Kingdoms represented: Data not present
##
## First 10 rows of data (use n = to show more):
##
## # A tibble: 1,119,509 x 14
##    yearMonth cellCode     basisofrecord      datasetkey     countrycode taxonKey
##    <chr>     <chr>        <chr>              <chr>          <chr>          <dbl>
##  1 1900-09   10kmE401N256 PRESERVED_SPECIMEN 98436813-7443~ CH           2394486
##  2 1900-09   10kmE402N257 PRESERVED_SPECIMEN 98436813-7443~ CH           2394486
##  3 1900-08   10kmE358N280 PRESERVED_SPECIMEN 15f819bd-6612~ FR           3190653
##  4 1900-08   10kmE359N282 PRESERVED_SPECIMEN 15f819bd-6612~ FR           3190653
##  5 1900-08   10kmE403N225 PRESERVED_SPECIMEN 0d8cc344-d880~ FR           3190653
##  6 1900-07   10kmE389N266 OBSERVATION        857aa892-f762~ FR           3190653
##  7 1900-07   10kmE390N266 HUMAN_OBSERVATION  dd238f50-f594~ FR           3190653
##  8 1900-07   10kmE390N267 HUMAN_OBSERVATION  2700f1e5-0f50~ FR           3190653
##  9 1900-07   10kmE402N257 PRESERVED_SPECIMEN 98436813-7443~ CH           2394486
## 10 1900-07   10kmE435N219 PRESERVED_SPECIMEN 5445131a-b491~ IT           3190653
## # i 1,119,499 more rows
## # i 8 more variables: scientificName <chr>, family <chr>, class <chr>,
## #   obs <dbl>, year <dbl>, xcoord <dbl>, ycoord <dbl>, resolution <chr>
```