Data mobilisation from GBIF to the EBV Data Portal for the Birds Directive Annex I

Notebook 01 - Prototype for creating a occurrence cube of birds listed in the Annex I of the Birds Directive

true

2024-08-27

Introduction

In this notebook we create a cube of occurrence of birds listed in the Annex I of the Birds Directive based on all species records available in GBIF until July 2024. To do this, we sent a request with the cube specifications (JSON file) via the GBIF API. We used the occurrence cube generation software developed in the Biodiversity Building Blocks for policy project. Details of the GBIF data query are available at DOI 10.15468/dl.uh84tp.

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

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

```
##          used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells 2364576 126.3   23468820 1253.4  23032897 1230.1
## Vcells 4763608  36.4  698522256 5329.4 909436593 6938.5
```

```r
# 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 (list of species in the Birds Directive Annex I), 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 1km 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/birds/")

# Load the JSON with the query specifications
query_ias <- "query_birds_annex1_acceptedusagekeys.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/", 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 <- "0062979-240626123714530"  # respose identifier
```

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/0062979-240626123714530"
```

```
# 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 code of the 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"         "order_"        "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: 1575 - 2023
## Single-resolution cube with cell size 10km ^2
## Number of cells: 83077
## Grid reference system: eea
## Coordinate range:
##    xmin    xmax    ymin    ymax
## -161000  817000  -83000  844000
##
## Total number of observations: 43093587
## Number of species represented: 168
## Number of families represented: 43
##
## Kingdoms represented: Data not present
##
## First 10 rows of data (use n = to show more):
##
## # A tibble: 15,980,116 x 15
##    yearMonth cellCode     basisofrecord      datasetkey     countrycode taxonKey
##    <chr>     <chr>        <chr>              <chr>          <chr>          <dbl>
##  1 1575-10   10kmE393N322 HUMAN_OBSERVATION  8a863029-f435~ NL           2481912
##  2 1670-03   10kmE470N396 HUMAN_OBSERVATION  38b4c89f-584c~ SE           2481959
##  3 1694-12   10kmE464N404 HUMAN_OBSERVATION  38b4c89f-584c~ SE           2498347
##  4 1700-01   10kmE477N404 HUMAN_OBSERVATION  38b4c89f-584c~ SE           2475365
##  5 1730-10   10kmE446N354 PRESERVED_SPECIMEN e5c5cad9-b987~ DK           9813242
##  6 1732-04   10kmE395N328 HUMAN_OBSERVATION  8a863029-f435~ NL           2481912
##  7 1742-09   10kmE441N389 HUMAN_OBSERVATION  38b4c89f-584c~ SE           2498347
##  8 1743-05   10kmE389N315 HUMAN_OBSERVATION  8a863029-f435~ NL           2481912
##  9 1743-05   10kmE437N401 HUMAN_OBSERVATION  b124e1e0-4755~ NO           2473577
## 10 1744-05   10kmE439N403 HUMAN_OBSERVATION  b124e1e0-4755~ NO           2473577
## # i 15,980,106 more rows
## # i 9 more variables: scientificName <chr>, family <chr>, class <chr>,
## #   order_ <chr>, obs <dbl>, year <dbl>, xcoord <dbl>, ycoord <dbl>,
## #   resolution <chr>
```