dev\_to\_EBD\_format

N.M. Tarr

2023-08-30

knitr::opts\_chunk$set(  
 echo = TRUE,  
 warning = FALSE,  
 message = FALSE)  
knitr::opts\_knit$set(root.dir = '~/Documents/NCBA/species/')

# Purpose

Whereas the primary objective for this repository of code is to support the exploration and analysis of data by NC Bird Atlas staff, we are making an effort to support deployment of the code by other states as well. The NCBA has its own database, the Atlas Cache, which uses a different database schema than the Ebird Basic Dataset. Therefore, a function for wrangling NCBA data into a format that is consistent with the EBD would be valuable. Functions that use observation and checklist records as inputs could then be used by other states. This document details the format conversion function for making Atlas Cache records match the format of EBD records.

# Function

to\_EBD\_format <- function(dataframe, drop = FALSE) {  
 # Reformat columns to match that of the EBD  
 #  
 # Description:  
 # Change the column names of a data frame retrieved from the NCBA Atlas Cache   
 # database to the format of tables retrieved from the eBird EBD. Also   
 # assign data types of columns to match those of EBD as read in by the auk   
 # package. This functions first assesses whether a data frame has the EBD  
 # format. If not, it then reformats the data frame to match the EBD. It   
 # then checks again for compliance before returning output.  
 #   
 # Parameters:  
 # dataframe -- a data frame retrieved from the NCBA Atlas Cache.  
 # drop -- TRUE or FALSE whether to drop columns not present in the EBD.  
  
 # Column capitalization ------------------------------------------------------  
 names(dataframe) <- str\_to\_lower(names(dataframe))  
   
 # Column exclusion (dropping) ------------------------------------------------  
 # Some names are in the EBD but not the example data set from auk, add them.  
 EBD\_names <- EBD\_fields(case = "lower")  
   
 # Drop extra field  
 if (drop == TRUE) {  
 dataframe <- dataframe %>% select(any\_of(EBD\_names))  
 }  
  
 # Transform may work when drop = FALSE but not TRUE bcs. of extra fields.  
 # Data types -----------------------------------------------------------------  
 # Transform the data types of the columns in records that are also in EBD and  
 # have different data types.  
 df2 <- transform(dataframe, bcr\_code = as.integer(bcr\_code),  
 duration\_minutes = as.integer(duration\_minutes),  
 effort\_area\_ha = as.numeric(effort\_area\_ha),  
 all\_species\_reported = as.logical(all\_species\_reported),  
 exotic\_code = as.logical(exotic\_code),  
 observation\_date = as.Date(observation\_date),  
 observation\_count = as.character(observation\_count),  
 has\_media = as.logical(has\_media)#,  
 #taxonomic\_order = as.numeric(taxonomic\_order),  
 #approved = as.logical(approved),  
 #reviewed = as.logical(reviewed)  
 )  
   
   
 return(df2)  
}

# Usage

Get a dataframe of observations from the Atlas Cache

# Load the tidyverse packages  
library(tidyverse)  
  
# Load the NCBA functions  
setwd("~/Code/NCBA/resources")  
source("ncba\_functions.R")  
  
# Point to and load the config file  
ncba\_config <- "~/Documents/NCBA/Scripts/ncba\_config.R"  
source(ncba\_config)  
  
# Get observations  
observations <- get\_observations(species = "King Rail", database = "AtlasCache",   
 NCBA\_only = TRUE, EBD\_fields\_only = FALSE,   
 fields = NULL, ncba\_config = ncba\_config)

Convert the dataframe to the EBD format.

observations2 <- to\_EBD\_format(observations, drop = TRUE)  
  
# Preview  
print(head(observations2))

## global\_unique\_identifier last\_edited\_date  
## 1 URN:CornellLabOfOrnithology:EBIRD:OBS1058111598 2022-01-05 17:21:09.562832  
## 2 URN:CornellLabOfOrnithology:EBIRD:OBS1058025708 2022-01-05 17:20:25.820528  
## 3 URN:CornellLabOfOrnithology:EBIRD:OBS1101358049 2022-04-29 20:47:22.136517  
## 4 URN:CornellLabOfOrnithology:EBIRD:OBS1107602423 2022-10-29 06:45:51.589976  
## 5 URN:CornellLabOfOrnithology:EBIRD:OBS1104665454 2022-10-29 06:45:51.589976  
## 6 URN:CornellLabOfOrnithology:EBIRD:OBS1104650137 2022-10-29 06:45:51.589976  
## taxonomic\_order category common\_name scientific\_name observation\_count  
## 1 5130 species King Rail Rallus elegans 1  
## 2 5130 species King Rail Rallus elegans 1  
## 3 5130 species King Rail Rallus elegans 2  
## 4 5187 species King Rail Rallus elegans 1  
## 5 5187 species King Rail Rallus elegans 1  
## 6 5187 species King Rail Rallus elegans 1  
## breeding\_code breeding\_category country country\_code state  
## 1 United States US North Carolina  
## 2 United States US North Carolina  
## 3 United States US North Carolina  
## 4 United States US North Carolina  
## 5 United States US North Carolina  
## 6 United States US North Carolina  
## state\_code county county\_code iba\_code bcr\_code usfws\_code atlas\_block  
## 1 US-NC Dare US-NC-055 US-NC\_402 27 USFWS\_80 35075F4NW  
## 2 US-NC Dare US-NC-055 US-NC\_323 27 35075G5CW  
## 3 US-NC Dare US-NC-055 US-NC\_323 27 35075B5NE  
## 4 US-NC Wake US-NC-183 US-NC\_314 29 35078F8NW  
## 5 US-NC Wake US-NC-183 US-NC\_314 29 35078F8NW  
## 6 US-NC Wake US-NC-183 US-NC\_314 29 35078F8NW  
## locality locality\_id  
## 1 Pea Island NWR--North Pond L130800  
## 2 Bodie Island Lighthouse & Pond L365371  
## 3 Cape Hatteras NS--Cape Point Campground L1448233  
## 4 American Tobacco Trail--New Hill-Olive Chapel Road Access L1475968  
## 5 American Tobacco Trail--New Hill-Olive Chapel Road Access L1475968  
## 6 American Tobacco Trail--New Hill-Olive Chapel Road Access L1475968  
## locality\_type latitude longitude observation\_date time\_observations\_started  
## 1 H 35.71758 -75.49858 2021-01-21 10:25:00  
## 2 H 35.81908 -75.56297 2021-01-20 15:30:00  
## 3 H 35.23556 -75.53884 2021-03-23 07:28:00  
## 4 H 35.71530 -78.94308 2021-03-27 10:45:00  
## 5 H 35.71530 -78.94308 2021-03-27 10:45:00  
## 6 H 35.71530 -78.94308 2021-03-27 08:30:00  
## observer\_id sampling\_event\_identifier protocol\_type protocol\_code  
## 1 obsr375303 S79785889 Traveling P22  
## 2 obsr375303 S79778982 Traveling P22  
## 3 obsr740338 S83935845 Traveling P22  
## 4 obsr1374383 S84432913 Traveling P22  
## 5 obsr981980 S84196664 Traveling P22  
## 6 obsr206965 S84195406 Traveling P22  
## project\_code duration\_minutes effort\_distance\_km effort\_area\_ha  
## 1 EBIRD\_ATL\_NC 195 4.828 0  
## 2 EBIRD\_ATL\_NC 60 0.402 0  
## 3 EBIRD\_ATL\_NC 59 2.855 0  
## 4 EBIRD\_ATL\_NC 119 2.889 0  
## 5 EBIRD\_ATL\_NC 119 2.889 0  
## 6 EBIRD\_ATL\_NC 260 2.012 0  
## number\_observers all\_species\_reported group\_identifier has\_media approved  
## 1 2 TRUE G6198731 FALSE TRUE  
## 2 2 TRUE G6198053 FALSE TRUE  
## 3 1 TRUE FALSE TRUE  
## 4 2 TRUE G6476590 FALSE TRUE  
## 5 2 TRUE G6476590 FALSE TRUE  
## 6 1 TRUE FALSE TRUE  
## reviewed reason  
## 1 FALSE   
## 2 FALSE   
## 3 FALSE   
## 4 TRUE   
## 5 TRUE   
## 6 TRUE   
## trip\_comments  
## 1 Mileage is walking & driving.  
## 2   
## 3   
## 4   
## 5   
## 6 Temp at 0830 hrs was 58 degrees F with fog and then cloudy skies. --Walked all the way down to Olive Church Rd.  
## species\_comments  
## 1   
## 2   
## 3 Heard calling life bird  
## 4 Heard with Jeff Blalock who confirmed the call  
## 5 Heard with Jeff Blalock who confirmed the call  
## 6 Continuing bird at this location responded to call. Heard two different times --calling. Heard calling the third time on his own, gave a much longer call and very close to the bridge.----A winter or migrant visitor  
## behavior\_code taxon\_concept\_id exotic\_code  
## 1 <NA> NA  
## 2 <NA> NA  
## 3 avibase-0BB84341 NA  
## 4 avibase-0BB84341 NA  
## 5 avibase-0BB84341 NA  
## 6 avibase-0BB84341 NA

# Tests

## Dropping columns

Test that columns are dropped appropriately. Refers to lists of EBD and non-EBD columns that can be returned from ncba\_functions.R. First get and test records from AtlasCache.

# Get observations from AtlasCache  
AC.obs.T <- get\_observations(species = "King Rail", database = "AtlasCache",   
 NCBA\_only = TRUE, EBD\_fields\_only = FALSE,   
 fields = NULL, ncba\_config = ncba\_config)

Test that non-EBD fields are dropped when drop is set to TRUE. If this chunk returns FALSE, then the test is passed.

# Reformat  
AC.obs.T.2 <- to\_EBD\_format(AC.obs.T, drop = TRUE)  
  
# Test if any non EBD fields are present  
print(unique(str\_to\_lower(nonEBD\_fields()) %in% names(AC.obs.T.2)))

## [1] FALSE

Also get records with EBD\_fields\_only set to FALSE and test that non EBD fields are dropped when drop is set to TRUE. If this chunk returns FALSE, then the test is passed.

# Get a data frame of AtlasCache records with ebd\_fields\_only set to false.  
AC.obs.F <- get\_observations(species = "King Rail", database = "AtlasCache",   
 NCBA\_only = TRUE, EBD\_fields\_only = FALSE,   
 fields = NULL, ncba\_config = ncba\_config)  
  
# Reformat  
AC.obs.F.2 <- to\_EBD\_format(AC.obs.F, drop = TRUE)  
  
# Test if any non EBD fields are present  
print(unique(str\_to\_lower(nonEBD\_fields()) %in% names(AC.obs.F.2)))

## [1] FALSE

Also make sure the test is passed with EBD records. Failure of this test would indicate that something is very wrong.

# Get observations from EBD  
EBD.obs <- get\_observations(species = "King Rail", database = "EBD",   
 NCBA\_only = TRUE, EBD\_fields\_only = FALSE,   
 fields = NULL, ncba\_config = ncba\_config)

Test that non EBD fields are dropped when drop is set to TRUE. If this chunk returns FALSE, then the test is passed.

# Reformat  
EBD.obs.2 <- to\_EBD\_format(EBD.obs, drop = TRUE)  
  
# Test if any non EBD fields are present  
print(unique(str\_to\_lower(nonEBD\_fields()) %in% names(EBD.obs.2)))

## [1] FALSE

The records from EBD should have all and only the EBD columns, so this chunk shouldn’t return any column names.

print(setdiff(EBD\_fields(case = "lower"), names(EBD.obs)))

## character(0)

print(setdiff(names(EBD.obs.2), EBD\_fields(case = "lower")))

## character(0)

When the records from AtlasCache were requested with EBD\_fields\_only set to TRUE, the result should have only the EBD columns because we ran the to\_EBD\_format function, so this chunk shouldn’t return any column names.

print(setdiff(names(AC.obs.T.2), EBD\_fields(case = "lower")))

## character(0)

When the records from AtlasCache were requested with EBD\_fields\_only set to TRUE, the result should have only the EBD columns because we ran the to\_EBD\_format function, so this chunk shouldn’t return any column names.

print(setdiff(names(AC.obs.F.2), EBD\_fields(case = "lower")))

## character(0)

## Column Names

Also print the column names to demonstrate that they were made lowercase.

# Print AtlasCache record names  
print(head(names(AC.obs.T.2)))

## [1] "global\_unique\_identifier" "last\_edited\_date"   
## [3] "taxonomic\_order" "category"   
## [5] "common\_name" "scientific\_name"

Examine EBD record columns names. The printed names should be lowercase.

print(head(names(EBD.obs.2)))

## [1] "checklist\_id" "global\_unique\_identifier"  
## [3] "last\_edited\_date" "taxonomic\_order"   
## [5] "category" "common\_name"

## Data Types

Examine the data types. Compare the data types of the output data frame from a query of AtlasCache to that of a query of the EBD. This chunk shouldn’t return the names of any columns if the test is passed.

# Get the data types of the columns  
record.types.EBD <- sapply(EBD.obs.2, class)  
record.types.AC.T <- sapply(AC.obs.T.2, class)  
record.types.AC.F <- sapply(AC.obs.F.2, class)  
  
# Get the names of the columns in AtlasCache records that are also in EBD  
common.columns1T <- intersect(names(EBD.obs.2), names(AC.obs.T.2))  
common.columns1F <- intersect(names(EBD.obs.2), names(AC.obs.F.2))  
  
# With ebd\_fields\_only = TRUE  
# Go through common columns and look for data type mismatches.  
print("EBD versus AtlasCache w/ EBD only = TRUE")

## [1] "EBD versus AtlasCache w/ EBD only = TRUE"

for (x in common.columns1T) {  
 if (record.types.EBD[x] != record.types.AC.T[x]) {  
 print(x)  
 print(record.types.EBD[x])  
 print(record.types.AC.T[x])  
 }  
}  
  
# With ebd\_fields\_only = FALSE  
# Go through common columns and look for data type mismatches.  
print("EBD versus AtlasCache w/ EBD only = FALSE")

## [1] "EBD versus AtlasCache w/ EBD only = FALSE"

for (x in common.columns1F) {  
 if (record.types.EBD[x] != record.types.AC.F[x]) {  
 print(x)  
 print(record.types.EBD[x])  
 print(record.types.AC.F[x])  
 }  
}

# Speed

Describe runtimes for a data-poor species.

# Get a data frame of AtlasCache records with ebd\_fields\_only set to false.  
 df <- get\_observations(species = "King Rail", database = "AtlasCache",   
 NCBA\_only = TRUE, EBD\_fields\_only = FALSE,   
 fields = NULL, ncba\_config = ncba\_config)  
  
# Run the function 5 times and record the runtime  
time <- c()  
for (i in 1:5) {  
 time1 <- proc.time()  
   
 # Reformat  
 to\_EBD\_format(df, drop = TRUE)  
 t <- proc.time() - time1  
 time[i] <- t["elapsed"]  
}  
  
# Print the descriptive statistics  
print(summary(time))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.001 0.002 0.002 0.002 0.002 0.003

Describe runtimes for a data-rich species.

# Get a data frame of AtlasCache records with ebd\_fields\_only set to false.  
 df <- get\_observations(species = "Indigo Bunting", database = "AtlasCache",   
 NCBA\_only = TRUE, EBD\_fields\_only = FALSE,   
 fields = NULL, ncba\_config = ncba\_config)  
  
# Run the function 5 times and record the runtime  
time <- c()  
for (i in 1:5) {  
 time1 <- proc.time()  
  
 # Reformat  
 to\_EBD\_format(df, drop = TRUE)  
 t <- proc.time() - time1  
 time[i] <- t["elapsed"]  
}  
  
# Print the descriptive statistics  
print(summary(time))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0890 0.0890 0.0900 0.0896 0.0900 0.0900