Butterfly_Checklist_Mobilization

Matthew Rogan

11/24/2021

Odonate Country Checklist Mobilization

Prepping Butterfly country checklists for integration with MOL

This script documents the MOL Country Checklist Mobilization workflow for the butterfly country checklists prepared by S. Pinkert and others. Is uses the general workflow detailed in the MOL_country_checklists Git Hub repository (https://github.com/matthewsrogan/MOL_country_checklists). A description of this workflow is also available from the MOL GDrive (https://drive.google.com/file/d/1HBCu9t6tkgtglLUA3boTmOuLobn9dkt0/view?usp=sharing).

The workflow consists of 7 steps: 1) Prep the R environment, 2) Read data, 3) Harmonize taxonomy, 4) Join spatial info, 5) Compute richness, 6) Run checks, and 7) Export checklist.

DATA DESCRIPTION The data for this checklist were provided by Stefan Pinkert (stefan.pinkert@yale.edu (mailto:stefan.pinkert@yale.edu)) and were prepared via a systematic literature review and GBIF occurrence records.

STEP 1: PREPARE R ENVIRONMENT

First we need to specify a few environmental variables. Begin by specifying the taxonomic group that the checklist represents.

```
taxa = "butterflies"

#create dataset short name - for reference purposes only
shortname = paste0(taxa, "_country_checklist")
```

Next we're going to specify source folders for data and the relevant code. Within the code directory should be the latest version of the MOL_country_checklists repository. We will also specify a root folder for all country checklist data (the migration directory).

```
#location of cloned Github MOL_country_checklists repository
pckgDir = "C:/Users/mr2577/Dropbox/BGC/MoL/Code/Country_checklists/MOL_country_checklists"

#migration directory - parent directory for processing regional checklists
migrationDir <- "G:/Shared drives/MOL_Data/Data/datasets/country_checklists"
```

We will also load the tidyverse suite of packages and source the custom functions from the repository.

We will also specify the names of the raw data files in preparation for reading them into the environment.

```
# Checklist data files
ccl_name <- "BttflyCCL_long_format(final).csv"

# Butterfly taxonomy
# use ValidBinomial as the accepted column for harmonization
taxo <- read_csv("G:/Shared drives/JetzLab/Specialty Groups/Groups/Taxon Expert Group/synonyms l
ists/master_taxo_with_syns/2021/masterTax_Butterflies_2021-11-20(long_format).csv")
```

```
## Rows: 113750 Columns: 18
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (16): sci_name, ValidName, ValidNameDate, Status, Family, Subfamily, Tri...
## dbl  (2): ValidNameID, Year
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

We are now ready to work with the data.

STEP 2: READ DATA INTO R

We will feed the file path to the raw data into the *read_checklist()* function. this function imports the data using *readr::read_csv()*. It then standardizes column names, runs checks on the data, and prints a summary. In addition to the file path, the function takes as arguments the names of relevant columns in the raw data. **NB: >14k records have 'NA' ISO3 codes.**

```
#path for checklist
rawPath <- file.path(migrationDir, taxa,  "in_prep/inputs/v2_112021", ccl_name)

#read in country checklist (ccl) using read_checklist
ccl <- read_ccl(rawPath = rawPath,
                geo_col = "Alpha.3",           #column name with ISO3 codes
                species_col = "ValidBinomial", #column name for species scientific names
                source_col = "source",         #column name for sources (optional)
                cols2keep = "supported_by")    #additional columns to keep from the source file
```

```
## [1] "This country checklist includes 8320406 observations of 19191 species across 249 countri
es."
```

```
## # A tibble: 6 x 4
##   scientificname        iso3  source    supported_by
##   <chr>                 <chr> <chr>     <chr>
## 1 Acraea andromacha     ASM   AcotboMMP ccl_only
## 2 Acytolepis puspa      FSM   AcotboMMP ccl_only
## 3 Appias ada            FSM   AcotboMMP both
## 4 Badamia exclamationis FSM   AcotboMMP ccl_only
## 5 Badamia exclamationis ASM   AcotboMMP ccl_only
## 6 Belenois java         ASM   AcotboMMP ccl_only
```

The dataset includes some observations that are based on GBIF records but that are not confirmed in the literature. We are going to filter out observations listed with “gbif” for the source.

```
ccl <- ccl %>%
  filter(source != "gbif")
```

STEP 3: CHECK TAXONOMY

Not all species in the checklist match preferred, accepted names. See a butterfly taxonomy comparison (https://drive.google.com/file/d/1Hz-uyKWSN7BeQNvYtftJjXLH49Tjnm7Z/view?usp=sharing) for more information about various discrepancies in the taxonomy. Some of the species in the checklist have what appear to be orthographic errors. One is considered ambiguous.

```
ccl_hrmnzd <- ccl_taxo(ccl = ccl, #the checklist
                       species_col = "scientificname", #column of species names in ccl
                       synlist = taxo, #master taxonomy in long format
                       canonical = "sci_name", #column with all canonicals, both accepted and sy
nonyms
                       harmonize = TRUE, #whether to merge column with accepted names to ccl
                       accepted_col = "ValidBinomial") #column of accepted names
```

```
## [1] "No unmatched canonicals."
```

```
rm(taxo)
```

The checklist also contains repeat observations of the same species-country pairs but from different sources. We’ll collapse these repeat observations.

```
ccl_updtd <- ccl_hrmnzd %>%
  distinct() %>%
  group_by(scientificname, iso3) %>%
  summarise(source = str_c(unique(source), collapse = "; "),
            verbatimscientificname = str_c(unique(verbatimscientificname), collapse = "; "),
            .groups ="drop")
```

STEP 4: JOIN GEOM IDs

Now we match ISO3 codes from the checklist to the MOL Geom IDs corresponding to each country using the *ccl_geoms()* function. The function takes as arguments the checklist containing an “iso3” column (e.g., the output of *read_checklist()*), the source folder for the repository (i.e., *pckgDir* from Step 1), and the logical argument “unmatch_fail” which determines how to handle records that could not be matched to MOL Geoms. The dataset has some ISO3 codes that appear to have typos and so the function will return unmatched records. Therefore, we will set *unmatch_fail = FALSE* and just extract the dataframe of matched records.

```
ccl_geo <- ccl_geoms(ccl_updtd,
                     pckgDir = pckgDir,
                     unmatch_fail = TRUE)
```

```
## Rows: 255 Columns: 2
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): iso3
## dbl (1): geomid
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

STEP 5: COMPUTE RICHNESS

Now we are ready to calculate country richness. We do this with the *ccl_richness()* function that accepts as arguments the ccl with accepted country codes, the name of the column that dictate species, and the taxa to be analyzed. Richness is calculated for each MOL geomid so it is necessary to run *ccl_geoms()* prior to performing this step. After running the function, we can save the output to a “ready_to_upload” folder within the migration folder (we’ll create it if it doesn’t exist).

```
richness <- ccl_richness(ccl_geo,
                         species_col = "scientificname",
                         taxa = taxa)
```

```
## [1] "butterflies country richness ranged between 1 and 3755 species per country."
```

```
#specify ready_to_upload folder
outDir <- file.path(migrationDir, taxa, "ready_to_upload")
if(!dir.exists(outDir)) dir.create(outDir)

#write a csv to the ready_to_upload
richness %>%
  write_csv(file.path(outDir, paste0(taxa, "_country_richness_v2.csv")))
```

STEP 6: DATA CHECKS

We’ll now run a couple of quick checks on the data to make sure everything looks in order. We’ll check for odd characters in the canonical. To do this, we’ll use the *check_odd_chr_canonical()* function from the taxonomy repository (copied to this repository for use here). We’ll also check the number of species and the number of countries. We’ll also make sure there aren’t any NAs lurking in the data.

```
#check canonicals
odd_chr <- check_odd_chr_canonical(ccl_geo, scientificname)
```

```
## [1] "No odd characters in canonical"
```

```
if(nrow(odd_chr) == 0) rm(odd_chr)

#check NAs in the geomid
sum(is.na(ccl_geo$geomid))
```

```
## [1] 0
```

```
#check NAs in the country
sum(is.na(ccl_geo$iso3))
```

```
## [1] 0
```

```
#get some counts
n_distinct(ccl_geo$geomid)
```

```
## [1] 244
```

```
n_distinct(ccl_geo$iso3)
```

```
## [1] 244
```

```
n_distinct(ccl_geo$scientificname)
```

```
## [1] 19138
```

```
n_distinct(ccl_geo$verbatimscientificname)
```

```
## [1] 19138
```

STEP 7: EXPORT CHECKLIST

We can use the *ccl_write()* function to export the data. It ensures columns have correct names and order. We’re also going to remove the *acceptedscientificname* column because the taxonomy will be harmonized within MOL.

```
ccl_write(ccl_geo,
          directory = outDir,
          geo_col = "geomid",
          species_col = "scientificname",
          country_col = "iso3",
          cols2keep = c("verbatimscientificname", "source"))
```

The odonate country checklists are prepped and ready for upload!