MOL_CountryChecklist_Mobilization

Matthew Rogan

10/25/2021

OVERVIEW

This document explains the process for prepping country checklists for a given taxa as inputs to the Map of Life (MOL). It prepares the data as species-country pairs and calculates richness within each country. It then matches the data for each country to MOL geom IDs for each ISO3 code. The process is run using a set of custom functions developed as the MOL_country_checklists repository on Git Hub (https://github.com/matthewsrogan/MOL_country_checklists). Contact Matt Rogan (matthew.rogan@yale.edu (mailto:matthew.rogan@yale.edu)) with any questions.

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

If you haven't already, complete the metadata form (https://docs.google.com/forms/d/1SEWbbNgdVzGN_YRAaO8sRl6kWTHRBGeMM_zt09sy-gk/edit) to ensure that your data are handled properly.

DATA DESCRIPTION

This workflow assumes that the raw data are provided in long format as a CSV such that each row in the CSV represents a single species-country association. Ideally, country info should be provided using ISO3 codes or using names that are directly relatable to ISO3 codes. The workflow can also incorporate a column that details the source for each record. The input files should be organized according to the standard drive structure (https://docs.google.com/drawings/d/1SUWRIhas3XIJx7Cf1a5LTOkRUvb57a6nUXBmSTWOVRk/edit?usp=sharing).

If a synonym list is provided, the workflow can also harmonize taxonomies.

STEP 1: PREPARE R ENVIRONMENT

First we need to specify a few environmental variables. Begin by specifying the taxonomic group that the checklist represents. We'll also create a dataset shortname to use when saving files.

```
#for odonates
#taxa = "dragonflies"

#for butterflies
taxa = "butterflies"

#create dataset short name
shortname = paste0(taxa, "_country_checklist")
```

Next we're going to specify source folders for data and the relevant code. We need to specify the folder with 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 country checklists
migrationDir <- "G:/Shared drives/MOL/Data/datasets/country_checklists"
```

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

Then specify the names of the raw data files in preparation for reading them into the environment.

```
# Checklist data files
#ccl_name <- "MOLOdonateChecklistData_0821.csv" #For dragonflies

ccl_name <- "unique_combinations_(072021, literature and gbif).csv" #For butterflies

#synonym list
#synList <- googlesheets4::read_sheet("https://docs.google.com/spreadsheets/d/1_WSXtJa3kD32VxHeCKhSe6ovLtWgO7AuEsFpFoj61Fo/edit#gid=306640760")

synList <- read_csv("G:/Shared drives/JetzLab/Specialty Groups/Groups/Taxon Expert Group/synonyms lists/master_taxo_with_syns/2021/masterTax_Butterflies_2021-09-07(long_format).csv")
```

```
## Rows: 113738 Columns: 17
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (15): 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.

```
#path for checklist
rawPath <- file.path(migrationDir, taxa, ccl_name)

#read in country checklist (ccl) using read_checklist
ccl <- read_ccl(rawPath = rawPath,
                geo_col = "GID_0",             #column name with ISO3 codes
                species_col = "ValidBinomial", #column name for species scientific names
                source_col = NULL)             #column name for sources (optional)
```

```
## New names:
## * `` -> ...1
```

```
## Warning in read_ccl(rawPath = rawPath, geo_col = "GID_0", species_col =
## "ValidBinomial", : 86 observations are missing geographic IDs.
```

```
## [1] "This country checklist includes 95742 observations of 18587 species across 257 countrie
s."
```

```
## # A tibble: 6 x 2
##   scientificname  iso3
##   <chr>           <chr>
## 1 Eurema mexicana MEX
## 2 Abaeis nicippe  BHS
## 3 Abaeis nicippe  HTI
## 4 Abaeis nicippe  CUB
## 5 Abaeis nicippe  JEY
## 6 Abaeis nicippe  CRI
```

STEP 3: CHECK TAXONOMY AND HARMONIZE IF NECESSARY

This step is mainly a form of quality control but can be used to match accepted scientific names to canonicals that should not count towards species richness (e.g., subspecies). The synonym list should be provided as a dataframe/tibble in long format where each row represents a canonical and its associated accepted scientific name. Ambiguous synonyms should be removed unless these synonyms are intended to be double counted. If no harmonization is being done, the synonym list can be a csv with a single column of valid canonicals.

```
ccl_taxo_check <- ccl_taxo(ccl          = ccl, #country checklist tibble
                           species_col  = "scientificname", #name of column with canonicals in c
cl
                           synlist      = synList, #master taxonomy
                           canonical    = "ValidName", #name of column with canonicals in synlis
t
                           harmonize    = FALSE, #if true, also harmonizes
                           accepted_col = NULL) #column of accepted canonicals to merge
```

```
## Warning in ccl_taxo(ccl = ccl, species_col = "scientificname", synlist =
## synList, : 28 canonicals are not listed in the synonym list.
```

```
#if there are some invalid taxa, they can be filtered out
ccl <- ccl %>%
  filter(!scientificname %in% ccl_taxo_check$scientificname)
```

If there are other cleaning steps that also need to be run, that can be done now. For example, the odonate literature lists some species from the Dutch Antilles, which now constitutes three separate countries. In the case of butterflies, some former French colonies were tagged with “FRA” instead of their current country codes. To fix

these problems, we can just execute ad hoc scripts to replace the values as needed.

```
# For odonates
# see odonate_country_checklist_mobilization.html

# For butterflies
source(file.path(pckgDir, "scripts/butterflies_fix_ISO3.R"))
```

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. If *unmatch_fail = TRUE* (the default) and not all records match, the function returns the unmatched records as a tibble. if *unmatched_fail = FALSE* and not all records match, the function returns a list of two tibbles, the first consisting of all matched records and the second consisting of all unmatched records. If all records match, then the function returns just one tibble.

```
ccl_geo <- ccl_geoms(ccl,
                     pckgDir = pckgDir)
```

```
## 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.
```

```
## Warning in ccl_geoms(ccl, pckgDir = pckgDir): 5 records in the country checklist
## do not have valid ISO3 codes. Returning tibble of unmatched records.
```

Notice in the case of butterflies that the function returns a warning that 6 records did not have valid ISO3 codes. Because we accepted the default argument of *unmatch_fail = TRUE*, the function only returned a tibble with those problematic records. We can look at those values:

```
## # A tibble: 5 x 2
##   scientificname         iso3
##   <chr>                  <chr>
## 1 Ithomiola nepos        BRZ
## 2 Mycalesis anapita      BRU
## 3 Neptis ochracea        XCA
## 4 Perisama oppelii       XCA
## 5 Pseudoneorina coulleti fossil
```

The fossil we can discard. “XCA” refers to the Caspian Sea so we won’t associate it with any particular country. “BRZ,” “BRU,” and “SPN” are probably typos, so we will have to ignore them for now. Since we can’t fix these cases, let’s rerun *ccl_geoms()* but set *unmatch_fail = FALSE*. Then, because we don’t care about these problematic records anymore, we’ll extract just the tibble of matched records.

```
ccl_geo <- ccl_geoms(ccl,
                     pckgDir = pckgDir,
                     unmatch_fail = FALSE)
```

```
## Warning in ccl_geoms(ccl, pckgDir = pckgDir, unmatch_fail = FALSE): 5 records in
## the country checklist do not have valid ISO3 codes. Returning a list of length
## two containing a tibble of matched records and a tibble of unmatched records.
```

```
#extract only tibble of matched records
ccl_geo <- ccl_geo[["matched"]] #here, "unmatch" would return the tibble of 6 problematic record
s
```

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 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 3209 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(paste0(outDir, "/country_richness.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"
```

```
#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] 249
```

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

```
## [1] 249
```

STEP 7: EXPORT CHECKLIST

We can use the *ccl_write()* function to export the data. It ensures columns have correct names and order. It expects three column names (can be left as default options if following this workflow closely) but additional colums (e.g., a 'source' or 'verbatimscientficname') can be included as a vector of column names fed to the *cols2keep* argument.

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

Congratulations! Your country checklist is now ready for integration with the Map of Life. One last reminder that if you haven't already, complete the metadata form (https://docs.google.com/forms/d/1SEWbbNgdVzGN_YRAaO8sRl6kWTHRBGeMM_zt09sy-gk/edit) to ensure that your data are handled properly.