**Introduction**

We have observed a tremendous rise in the availability of data in all fields of biodiversity research and management during the last decade. Most of these new data sources are scattered and not connected among each other. The diversity of research and management questions to be answered requires a flexible but also standardised way of connecting and merging individual data sets. This is commonly done using individual workflows specific for a certain project using very different standards and practices. Unfortunately, this leads to the situation that the obtain products are not comparable among each other and consequently that the procedures of standardisation and merging have to be developed repeatedly for individual projects. Using standardised publicly available workflows would allow generating combined databases in a transparent and repeatable way.

We here introduce a workflow to combine different databases of spatial and temporal information of alien species occurrences. These databases have been often established for specific taxonomic groups or environments or with a focus on certain species characteristics (e.g. invasive). This, however, necessitates merging these databases, which have been done in number of studies. Although the procedure of combining the data sets were described in the respective publication, the comparable low degree of information and the multitude of decisions, which have been taken by the developers, impede transparency and repeatability. A standardised, published and publicly accessible workflow would have allowed repeating and refining the process without repeating the time-consuming procedure of developing a new workflow. Although the workflow was developed for the merging of databases of alien species occurrences, it can be easily adapted to any other database including spatial information of biological data.

**Overview**

The workflow is a fully automated process implemented in the statistical software R. The workflow handles information, which is organised as single tables with a minimum content of species names and region names. The user has to enter summary characteristics of the respective databases into a configuration table, which represents the basis for further processing. The workflow consists of four steps, namely 1.) preparation of databases, 2.) standardisation of species names, 3.) standardisation of region names and 4.) merging of standardised databases. In the preparation phase, the databases are loaded and the required column names are identified and standardised. The standardisation of species names is done using the GBIF backbone taxonomy. Species names identified as synonyms according to GBIF are replaced with the accepted name in GBIF. In the third step, region names are standardised using a data set of spatial information. Region delineation is based upon ISO 3166-1 alpha-2 codes with a few additional entries, which are important for alien species. The region table can be modified by the user. In the final step, all standardised databases are merged based on information of species and region names. Duplicated entries are removed. Information deviating among databases is merged with a notification for further checking.

The workflow is easily accessible, fully transparent and offers a large flexibility to let the user decide how individual information is treated, how to standardise information and which information to add. The workflow exports mismatches, which easily allows the user to identify and potentially correct the provided information. It exports intermediate results, which enables the user to trace back single errors, mismatches of species names and region names, a full list of species (species level) including taxonomic information and the final databases including all information. For more complex procedures of the standardisation of region names, the user can provide individual R scripts, which will be incorporated in the workflow if provided. Each step of the workflow can be run individually if the output of the previous steps is provided. Original names of species and regions are kept for reference.

**Methods**

**Requirements**:

1. **R** installed on a computer. Running the work flow requires two R packages (“rgbif” and “openxlsx”), which can be installed in R executing the command “install.packages(c(“rgbif”,”openxlsx”)).
2. **Standard folder structure**: All files have to be stored in a folder with the subfolders “R/” (all R scripts), “Config/” (configuration file), “Inputfiles/” (data sets of alien species information). The workflow will generate an additional subfolder “Output/”, where all output is stored.
3. **R scripts of workflow, region table and sub-species identifier**
   1. R scripts include all scripts to run the workflow and optionally additional scripts to standardise region names, which can be provided by the user. All R scripts have to be stored in the subfolder “R/”.
   2. The region table (“AllRegionsList.xlsx”) contains information about the regions including a set of keywords, which enables the identification of region spelled in different ways. The region table has to be stored in the subfolder “Inputfiles/”.
   3. The table “SubspecIdentifier.xlsx” contains keywords describing information below the species level as part of the species names (e.g., “subsp.”). These keywords are used to remove information below the species level, which otherwise may lead to mismatches in GBIF (see below for more details). The table is stored in the subfolder “Inputfiles/”.
4. **Databases of alien species**: Databases have to be provided as Excel files (only \*.xlsx) with all required information included on the first sheet, which is formatted as a table. That is, the information has to be provided on the first sheet of the Excel file and has to be strictly formatted as a table (first line (single row): column headers; following lines: table rows; no empty lines; only first sheet will be read in). All databases have to be stored in the subfolder “Inputfiles/”.
5. **Configuration file**: The configuration file provides information about each data set of alien species. In this file the column headers in the original files are listed, which are used to extract the relevant information necessary to run the workflow. Note that only databases will be considered in the workflow, which are added to the configuration file. It represents an Excel file called “DatabaseInfo.xlsx” and is stored in the folder “Config/”. Information stored in the configuration file consists of three categories: essential, optional and additional:
   1. **Required columns**: Information in these columns isessential for further processing. It contains short names of databases, file names, species names and region names.
   2. **Optional columns**: This information can be used in the workflow if provided, but is not required to successfully run the workflow. For instance, information about the taxonomic group covered in the database can be used to check multiple entries in GBIF, which would be otherwise exported as missing matches of species names. Optional columns contain taxonomic group (currently included: “All”, “Amphibians”, “Reptiles”, “Birds”, “Vascular plants”), author, kingdom, country codes, first records (start and end, or only start; numeric) and invasion status.
   3. **Additional columns**: These columns are kept as they are provided in the original database without affecting the workflow. Columns names should be provided in a single column separated by “; ”.
6. Internet connection to access GBIF.

**Workflow**

The workflow consists of a series of four steps described below in detail. It can be executed by running the R script “runWorkflow.r” in the “R/” subfolder. Before doing so, the working directory has to be set within this file by setting the path to the working directory in the command “setwd()”. The working directory should be the main folder including all subfolders. Running the workflow will generate a new folder called “Output/”, which will be used to store all intermediate and final results of the workflow.

1. **Load and prepare individual databases** (R script “PrepareDatasets.r”):Load and prepare individual data sets of alien species based on the information provided in DatabaseInfo.xlsx. Individual steps:
   1. Check for column names and if provided rename column names consistently
   2. If species names and author names are provided in different columns, these columns are merged into a single one.
   3. First records are formatted as numeric values, which may generate missing values if a first record does not consists only of numeric values (e.g. “<1930”). In such a case, values will be transformed into missing values and a warning pops up indicating that the user should check the column entries of the respective database.
   4. Export all data sets with standardised column names
2. **Standardisation of species names** (R script “StandardiseSpeciesNames.r”): Compare species names with theGBIF backbone taxonomy (using name\_backbone() from R package rgbif) and report species name, scientific name (incl. author), status in GBIF (accepted or synonym) and taxonomic tree. Individual steps:
   1. ~~Remove all information below species level. This is done by reading in a user-defined file named “SubspecIdentifier.xlsx”, which contains keywords to identify any information in the species names belonging to descriptions below the species level. It is assumed that the keywords are part of the species name either ending with or without a dot with leading and trailing space (e.g., “ subsp. “ or “ subsp ”). Everything on the right side of the keyword is removed, which may include also author names. Author names will be added at a later stage (see below).~~
   2. Rules of name acceptance (R script “CheckGBIFTax.r”)
      1. Get exact match of scientific name (ie. Status=”accepted”, matchType=”exact” and extract species names (can be stored under “species” or “canonicalName” in GBIF). Sub-species level information such as varieties, forms or subspecies is ignored.
         1. Get author information and status of taxonomic description only if exact match is on species level.
         2. If exact match is on sub-species level, extract author name and status from species level in a new request to GIBF now using species names only (Status=”accepted”, matchType=”exact”). For example, “Trifolium pratense var. sativum Schreb.” is transformed to “Trifolium pratense L.” in this way.
      2. If exact match is not available, check for synonyms (i.e., status=”synonym”, matchType=”exact” and existing species level information)
         1. Author names are not provided for synonyms in GBIF. Thus, use new species name (i.e., the synonym) for a new GBIF request to get author name of accepted species name; ensure correct selection by additionally using family name to avoid false reports due to multiple entries in GBIF.
      3. If exact match is not available for species names or synonyms, check ‘alternative’ names (due to e.g. multiple entries in GBIF as a plant and insect), following these rules:
         1. If an exact match of an accepted species names is found, take it together with status (‘accepted’).
            1. Due to potential multiple entries, author names need to be carefully selected: If information about kingdom was provided in the original data file, select name and author from the respective kingdom (assuming that name matching only occurs across kingdoms, probably incorrect?).
            2. If not available, use information about ‘taxon\_group’ to get information (needs to be hard coded; currently implemented for “Vascular plants”, “Reptiles”, “Amphibians” and “Birds”).
         2. If an exact match of a synonym is found in alternatives GBIF results, check whether family names are identical and if so take the synonym together with its status.
      4. If none of the above matches the requested species name, store mismatch for export. Note that only exact matches are accepted. Misspellings are not checked as these checks are prone to false matches.
   3. Export databases with standardised species names, a full list of species with taxonomic information and a missing species names table. Missing species represent species without a match in GBIF based on the rules defined above. Note that missing species are kept in the final database.
   4. **Replace and add taxonomic information** by user-specified information. The user can specify taxonomic information in an excel file (“UserDefinedSpeciesNames.xlsx”), which will replace the information included so far. This allows modifying species names obtained from GBIF. To replace species names, species names in the database and new species name have to be provided. Information about higher taxonomic levels will be replaced if provided in the excel file. Taxonomic information about higher taxonomic levels and/or authors can be added by entering the same species name for old and new version plus additional taxonomic information. In this case, the species name is kept, while the new taxonomic information is added.
3. **Standardisation country names** (R script “StandardiseCountryNames.r”): Load standardised list of regions (“AllRegionsList.xlsx”) and replace original names. Region list is based on 2-digit ISO code (ISO 3166-1 alpha-2). List contains information about continents, TDWG large regions, GADM country names, ISO3, keywords etc.
   1. Some notes for consideration related to ISO standard:
      1. A few additional regions are added as these represent important hot spots and/or are clearly separated from the mainland country. These regions have the same ISO2 as the mainland country (to be discussed): Lord Howe Island (Australia), Tasmania (Australia), Easter Island (Chile), islands of Galapagos (Ecuador), Balearic Islands (Spain), Canary Islands (Spain), Corsica (France), Sardinia (Italy), Sicily (Italy), Kermadec Islands (New Zealand), Madeira (Portugal), Alaska (USA), Hawaiian Islands (USA)
      2. Note: Due to these duplicated ISO codes, country matching is done by country names rather than by ISO codes.
      3. Note: Many islands with loose connections to the mainland country are already separated with an unique ISO2 such as Reunion, French Southern Territories, British Indian Ocean Territory, US Minor Outlying Islands
      4. Note: Some distinctions are surprisingly detailed such as Isle of Man, Jersey, Guernsey, Gibraltar, Vatican, which is probably not useful for us.
   2. Standardisation of region names is done as follows:
      1. Clean region names (removing inappropriate space and “ (the)”)
      2. First check if an R script is provided, which does the name standardisation. This option is implemented as in some cases, country names are provided in an awkward way in the original data set. To avoid lengthy coding, the workflow can be linked to a user-provided R script. The name of this script has to be provided in the configuration file. In the workflow, the script is automatically loaded and executed. An example is provided for the amphibian-reptile data set.
      3. If not available, match by exact match of provided country names.
      4. Check remaining mismatches by comparing the keywords provided in the country file. Keywords represent alternative spellings or names of the country.
   3. If no match was found, check the output file “MissingRegions\_....csv”. The following options remain:
      1. Add region to list of standardised names if the region is missing
      2. Add keyword to list of standardised names for the specific region to enable the algorithm to find the respective name separated by “;” (e.g., “USA” for “United States of America”)
      3. Do nothing if the region should not enter the final data set.
   4. Export translation table of country names and missing names
4. **Standardisation of first records** (R script: “GetFirstRecord.r”): Treat first records according to user-specified rules. Transform non-numeric first records to numeric values.
   1. Load list of replacements from excel file (“Guidelines\_FirstRecords.xlsx”), which includes the rules to treat first records and a list of replacements. Entries specified in the column “Entry” will be replaced by values in column “Replacement”. The replacement is exact, which means that exactly the sequence of symbols in the first column will be replaced with the symbols specified in the second column. Additional symbols in the same entry will be kept untouched. For example, the entry “< “ with replacement “” will transform “< 1960” to “1960”.
   2. Entries, which cannot be transformed into numeric values, generate a warning and are exported into the file “NonNumericFirstRecords\_[database].csv” to check. These values can be treated by adding new lines in Guidelines\_FirstRecords.xlsx.
5. **Merge all standardised databases**
   1. Merge based on species names and region names
   2. Solve discrepancies if appearing
      1. First record: If two different first records are provided by different databases, select the earliest first record.
      2. For other columns, merge column content and report discrepancy as a warning. These need to be addressed by the user, who may then modify the original data files.

**Open questions:**

* + - 1. How to deal with wrong or missing entries in GBIF? Should we strictly stick to GBIF, which means that corrections have to be done through GBIF, or should we offer the option to make changes manually directly in the workflow?
      2. Which regions should be considered? Standard classifications such as ISO do not consider region, which are important for our purpose (e.g. Hawaii not separated from USA). Right now additional regions can be entered by modifying entries in the file “AllRegionsList.xlsx”. Personally, I prefer this solution though it is not a standard.
      3. Right now, first records are not standardised at all as this is not required in our case, because GRIIS and FirstRecords reported already standardised values. Nevertheless, it may make sense to include a process to resolve first records also to report how we did it in the other two databases.
      4. Shall we consider invasion status? Maybe too challenging to be resolved.