parseGBIF Manual

Pablo Hendrigo Alves de Melo[[1]](#footnote-20)

Nadia Bystriakova[[2]](#footnote-22)

Alexandre Monro[[3]](#footnote-24)

2023-07-12

# parseGBIF Manual

The parseGBIF package is designed to repackage [Global Biodiversity Information Facility - GBIF](https://www.gbif.org/) species occurrence records into a format that optimises its use in further analyses. Currently occurrence records in GBIF can include several duplicate digital records, and in the case of vascular plants, for several physical duplicates of unique collection events (biological collections). parseGBIF aims to parse these records to a single, synthetic, record corresponding to a unique collection event to which a standardized scientific name is associated. It does so by providing tools to verify and standardize species scientific names, score the quality of both the naming of a record and of its associated spatial data, and to use those scores to synthesise and parse duplicate records into unique collection events. This Manual provides a brief introduction to parseGBIF, with more information available from Help pages accessed via the help fuction. We believe that this package will be of particular use for analyses of plant occurrence data.

## Installation

You can install the development version of parseGBIF from [GitHub](https://github.com/pablopains/parseGBIF). To install parseGBIF, run

devtools::install\_github("pablopains/parseGBIF")

Please site parseGBIF as:

print(citation("parseGBIF"), bibtex = FALSE)

## To cite package 'parseGBIF' in publications use:  
##   
## Melo P, Bystriakova N, Monro A (2023). "ParsGBIF: An R package for  
## parsing species occurrence records." \_Methods in Ecology and  
## Evolution\_, \*1\*(11), 1-11. doi:doi..... <https://doi.org/doi.....>.

## Example

**Getting species occurrence records from GBIF**

### 1. GBIF data preparation

#### 1.1. Getting occurrence data of the species records from GBIF

1.1.1. Access a registered account in [GBIF](gbif.org)

1.1.2. Filter occurrences using available fields, for instance:

* Basis of record: *Preserved specimen*
* Occurrence status: *present*
* Scientific name: *Botanical family name* (e.g. Achatocarpaceae) or **filter by other fields**

1.1.3. Request to download information in **DARWIN CORE ARCHIVE FORMAT**

1.1.4. Download compressed file and unzip downloaded file

1.1.5. Use the **occurrence.txt** file as input to the prepare\_gbif\_occurrence\_data(gbif\_occurrece\_file = ‘occurrence.txt’) function

#### 1.2. Preparing occurrence data downloaded from GBIF

To prepare occurrence data downloaded from GBIF to be used by parseGBIF functions, run prepare\_gbif\_occurrence\_data.

library(parseGBIF)  
   
occ\_file <- 'https://raw.githubusercontent.com/pablopains/parseGBIF/main/dataGBIF/Achatocarpaceae/occurrence.txt'  
   
occ <- parseGBIF::prepare\_gbif\_occurrence\_data(gbif\_occurrece\_file = occ\_file, columns = 'standard')  
   
head(occ)

## # A tibble: 6 × 55  
## Ctrl\_gbifID Ctrl\_bibliographicCitation Ctrl\_language Ctrl\_institutionCode   
## <dbl> <chr> <chr> <chr>   
## 1 931015583 <NA> es Universidad de Antioquia…  
## 2 931008720 <NA> es Universidad de Antioquia…  
## 3 931008021 <NA> es Universidad de Antioquia…  
## 4 931002305 <NA> es Universidad de Antioquia…  
## 5 931001019 <NA> es Universidad de Antioquia…  
## 6 930997693 <NA> es Universidad de Antioquia…  
## # ℹ 51 more variables: Ctrl\_collectionCode <chr>, Ctrl\_datasetName <chr>,  
## # Ctrl\_basisOfRecord <chr>, Ctrl\_informationWithheld <chr>,  
## # Ctrl\_dataGeneralizations <chr>, Ctrl\_occurrenceID <chr>,  
## # Ctrl\_catalogNumber <chr>, Ctrl\_recordNumber <chr>, Ctrl\_recordedBy <chr>,  
## # Ctrl\_georeferenceVerificationStatus <chr>, Ctrl\_occurrenceStatus <chr>,  
## # Ctrl\_eventDate <dttm>, Ctrl\_year <dbl>, Ctrl\_month <dbl>, Ctrl\_day <dbl>,  
## # Ctrl\_habitat <chr>, Ctrl\_fieldNotes <chr>, Ctrl\_eventRemarks <chr>, …

When parsing data, the user can choose to select “standard” or “all” fields (columns). The “standard” format has 55 data fields (columns), and the “all” format, 257 data fields (columns).

col\_standard <- parseGBIF::select\_gbif\_fields(columns = 'standard')  
   
str(col\_standard)

## chr [1:55] "gbifID" "bibliographicCitation" "language" "institutionCode" ...

col\_all <- parseGBIF::select\_gbif\_fields(columns = 'all')  
  
str(col\_all)

## chr [1:257] "gbifID" "abstract" "accessRights" "accrualMethod" ...

#### 1.3. Extracting GBIF issues

GBIF recognises and documents several issues relating to the data fields for an individual record. The issue field stores terms that represent an enumeration of GBIF validation rules. Issues can lead to errors or unexpected data. The issues fields are therefore a valuable source of information when assessing the quality of a record. In order to help GBIF and the data publishers improve the data, GBIF flag records with various issues that they have encountered. These issues can be used as filters applied to occurrence searches. Not all issues indicate bad data, some flagthe fact that GBIF has altered values during processing. The values of EnumOccurrenceIssue will be used by the function extract\_gbif\_issue as a model to tabulate the GBIF issues of each record, individualizing them, in columns.TRUE or FALSE, flagging whether the issue applies or not for each record.

data(EnumOccurrenceIssue)  
  
colnames(EnumOccurrenceIssue)

## [1] "constant" "description" "definition" "type"   
## [5] "priority" "score" "selection\_score" "reasoning"   
## [9] "notes"

head(dplyr::arrange(EnumOccurrenceIssue,desc(score)))

## # A tibble: 6 × 9  
## constant description definition type priority score selection\_score reasoning  
## <chr> <chr> <chr> <chr> <chr> <dbl> <dbl> <chr>   
## 1 COORDIN… Coordinate… A coordin… geos… High 3 -9 Records …  
## 2 COORDIN… Coordinate… The suppl… geos… High 3 -9 Records …  
## 3 COUNTRY… The interp… The inter… geos… High 3 -9 Records …  
## 4 ZERO\_CO… Coordinate… Coordinat… geos… High 3 -9 Records …  
## 5 COORDIN… The given … The given… geos… Medium 2 -3 Potentia…  
## 6 COORDIN… Indicates … Indicates… geos… Medium 2 -3 Potentia…  
## # ℹ 1 more variable: notes <chr>

occ\_gbif\_issue <- parseGBIF::extract\_gbif\_issue(occ = occ)  
  
names(occ\_gbif\_issue)

## [1] "occ\_gbif\_issue" "summary"

head(occ\_gbif\_issue$summary)

## issue n\_occ  
## 1 INSTITUTION\_MATCH\_FUZZY 1919  
## 2 GEODETIC\_DATUM\_ASSUMED\_WGS84 1883  
## 3 OCCURRENCE\_STATUS\_INFERRED\_FROM\_INDIVIDUAL\_COUNT 1336  
## 4 CONTINENT\_DERIVED\_FROM\_COORDINATES 1038  
## 5 COORDINATE\_ROUNDED 978  
## 6 TYPE\_STATUS\_INVALID 858

### 2. Check species names against WCVP database

The World Checklist of Vascular Plants (WCVP) database is available from the (Royal Botanic Gardens, Kew)[<https://powo.science.kew.org/about-wcvp>]. It can be downloaded to a folder of the user’s choice or into memory using get\_wcvp function. The output has 33 columns.

data(wcvp\_names\_Achatocarpaceae)  
wcvp\_names <- wcvp\_names\_Achatocarpaceae  
   
# wcvp\_names <- wcvp\_get\_data(read\_only\_to\_memory = TRUE)$wcvp\_names  
   
colnames(wcvp\_names)

## [1] "plant\_name\_id" "ipni\_id"   
## [3] "taxon\_rank" "taxon\_status"   
## [5] "family" "genus\_hybrid"   
## [7] "genus" "species\_hybrid"   
## [9] "species" "infraspecific\_rank"   
## [11] "infraspecies" "parenthetical\_author"   
## [13] "primary\_author" "publication\_author"   
## [15] "place\_of\_publication" "volume\_and\_page"   
## [17] "first\_published" "nomenclatural\_remarks"   
## [19] "geographic\_area" "lifeform\_description"   
## [21] "climate\_description" "taxon\_name"   
## [23] "taxon\_authors" "accepted\_plant\_name\_id"   
## [25] "basionym\_plant\_name\_id" "replaced\_synonym\_author"  
## [27] "homotypic\_synonym" "parent\_plant\_name\_id"   
## [29] "powo\_id" "hybrid\_formula"   
## [31] "reviewed" "TAXON\_NAME\_U"   
## [33] "TAXON\_AUTHORS\_U"

Species’ names can be checked against WCVP database one by one, or in a batch mode. To verify individual names, the function wcvp\_check\_name is used.

name.checked <- wcvp\_check\_name(searchedName = 'Achatocarpus mollis H.Walter',  
 wcvp\_names = wcvp\_names,  
 if\_author\_fails\_try\_without\_combinations = TRUE)  
name.checked[,c(3:5,22,23,40)]

## wcvp\_taxon\_rank wcvp\_taxon\_status wcvp\_family wcvp\_taxon\_name  
## 9 Species Accepted Achatocarpaceae Achatocarpus pubescens  
## wcvp\_taxon\_authors wcvp\_searchNotes  
## 9 C.H.Wright Updated

To check names in a batch mode, there is wcvp\_check\_name\_batch function. It uses the occurrence data (occ) and WCVP names list (wcvp\_names) generated in the previous steps.

names.checked <- wcvp\_check\_name\_batch(occ = occ,  
 wcvp\_names = wcvp\_names,  
 if\_author\_fails\_try\_without\_combinations = TRUE,  
 wcvp\_selected\_fields = 'standard')  
   
names(names.checked)

## [1] "occ\_wcvp\_check\_name" "summary"

head(names.checked$summary)

## wcvp\_plant\_name\_id wcvp\_taxon\_rank wcvp\_taxon\_status wcvp\_family  
## 4 500156 Species Accepted Achatocarpaceae  
## 22 500161 Form Accepted Achatocarpaceae  
## 15 500146 Species Accepted Achatocarpaceae  
## 9 500163 Species Accepted Achatocarpaceae  
## 2 500150 Species Accepted Achatocarpaceae  
## 16 500149 Species Accepted Achatocarpaceae  
## wcvp\_taxon\_name wcvp\_taxon\_authors  
## 4 Achatocarpus nigricans Triana  
## 22 Achatocarpus praecox f. obovatus (Schinz & Autran) Hauman  
## 15 Achatocarpus balansae Schinz & Autran  
## 9 Achatocarpus pubescens C.H.Wright  
## 2 Achatocarpus gracilis H.Walter  
## 16 Achatocarpus brevipedicellatus H.Walter  
## wcvp\_accepted\_plant\_name\_id wcvp\_reviewed  
## 4 500156 Y  
## 22 500161 Y  
## 15 500146 Y  
## 9 500163 Y  
## 2 500150 Y  
## 16 500149 Y  
## wcvp\_searchedName wcvp\_taxon\_status\_of\_searchedName  
## 4 Achatocarpus nigricans Triana <NA>  
## 22 Achatocarpus obovatus Schinz & Autran Synonym  
## 15 Achatocarpus balansae Schinz & Autran <NA>  
## 9 Achatocarpus pubescens C.H.Wright <NA>  
## 2 Achatocarpus gracilis H.Walter <NA>  
## 16 Achatocarpus brevipedicellatus H.Walter <NA>  
## wcvp\_plant\_name\_id\_of\_searchedName wcvp\_taxon\_authors\_of\_searchedName  
## 4 NA <NA>  
## 22 500158 Schinz & Autran  
## 15 NA <NA>  
## 9 NA <NA>  
## 2 NA <NA>  
## 16 NA <NA>  
## wcvp\_verified\_author wcvp\_verified\_speciesName wcvp\_searchNotes  
## 4 100 100 Accepted  
## 22 100 100 Updated  
## 15 100 100 Accepted  
## 9 100 100 Accepted  
## 2 100 100 Accepted  
## 16 100 100 Accepted

To bring species’ names into line with the format used by WCVP, the function standardize\_scientificName inserts a space between the hybrid separator (x) and specific epithet, and also standardizes abbreviations of infrataxa (variety, subspecies, form).

# hybrid separator  
standardize\_scientificName('Leucanthemum ×superbum (Bergmans ex J.W.Ingram) D.H.Kent')

## $searchedName  
## [1] "Leucanthemum ×superbum (Bergmans ex J.W.Ingram) D.H.Kent"  
##   
## $standardizeName  
## [1] "Leucanthemum × superbum"  
##   
## $taxonAuthors  
## [1] "(Bergmans ex J.W.Ingram) D.H.Kent"  
##   
## $taxonAuthors\_last  
## [1] "D.H.Kent"

# variety   
  
standardize\_scientificName('Urera baccifera var. angustifolia Wedd.')

## $searchedName  
## [1] "Urera baccifera var. angustifolia Wedd."  
##   
## $standardizeName  
## [1] "Urera baccifera var. angustifolia"  
##   
## $taxonAuthors  
## [1] "Wedd."  
##   
## $taxonAuthors\_last  
## [1] ""

# subspecies  
standardize\_scientificName('Platymiscium pubescens subsp. fragrans (Rusby) Klitg.')

## $searchedName  
## [1] "Platymiscium pubescens subsp. fragrans (Rusby) Klitg."  
##   
## $standardizeName  
## [1] "Platymiscium pubescens subsp. fragrans"  
##   
## $taxonAuthors  
## [1] "(Rusby) Klitg."  
##   
## $taxonAuthors\_last  
## [1] "Klitg."

The function collectors\_get\_name returns the last name of the main collector in recordedBy field. It standardizes the text string to replace non-ascii characters.

# library(parseGBIF)  
  
collectors\_get\_name('Müller, W.')

## [1] "Muller"

collectors\_get\_name("PEDRO ACEVEDO-RODRÍGUEZ|A. SIACA|GEORGE R. PROCTOR|JULIE F. BARCELONA|J.A. CEDEÑO|P. LEWIS|R. O'REILLY|E. SANTIAGO")

## [1] "ACEVEDO-RODRIGUEZ"

collectors\_get\_name("BORNMÜLLER, JOSEPH FRIEDRICH NICOLAUS")

## [1] "BORNMULLER"

collectors\_get\_name("Botão, S.R.; Machado, F.P.")

## [1] "Botao"

collectors\_get\_name('Melo, P.H.A, Bystriakova, N. & Monro, A.')

## [1] "Melo"

collectors\_get\_name('Monro, A.; Bystriakova, N. & Melo, P.H.A')

## [1] "Monro"

collectors\_get\_name('Bystriakova, N., Monro, A.,Melo, P.H.A')

## [1] "Bystriakova"

### 3. Collectors Dictionary

To extract the last name of the main collector based on the recordedBy field and assemble a list relating the last name of the main collector and the raw data from the recordedBy, use the collectors\_prepare\_dictionary function. It uses the occurrence data (occ) generated in the previous step.

#### 3.1 Prepare dictionary collectors

collectorsDictionary.dataset <- collectors\_prepare\_dictionary(occ = occ)  
  
head(collectorsDictionary.dataset)

## Ctrl\_nameRecordedBy\_Standard Ctrl\_recordedBy  
## 1 ACEVEDO R. ACEVEDO R.; C. REYES  
## 2 ACOSTA R. ACOSTA P.; F. VÁZQUEZ B.  
## 3 ACOSTA R. ACOSTA P.; N. ACOSTA B.  
## 4 ADARVE J. ADARVE  
## 5 AGUILAR AGUILAR, R.M.  
## 6 AGUILAR JOSÉ AGUILAR CANO;SANDRA MEDINA;JHONATAN GUZMÁN  
## Ctrl\_notes collectorDictionary Ctrl\_update collectorName Ctrl\_fullName  
## 1 <NA> <NA> <NA> <NA>  
## 2 <NA> <NA> <NA> <NA>  
## 3 <NA> <NA> <NA> <NA>  
## 4 <NA> <NA> <NA> <NA>  
## 5 <NA> <NA> <NA> <NA>  
## 6 <NA> <NA> <NA> <NA>  
## Ctrl\_fullNameII CVStarrVirtualHerbarium\_PersonDetails  
## 1 <NA> <NA>  
## 2 <NA> <NA>  
## 3 <NA> <NA>  
## 4 <NA> <NA>  
## 5 <NA> <NA>  
## 6 <NA> <NA>

#### 3.2 Check the main collector’s last name

It is recommended to check the main collector’s last name in the nameRecordedBy\_Standard field. Our goal is to standardize the main collector’s last name, which is automatically extracted from the recordedBy field. We do so by standardizing the text string so that all characters are replaced by uppercase and non-ascii characters, so that collector reponsible for a collection event is always recorded using the same string of characters.

If the searched recordedBy entry is present in the collector’s dictionary, the function retrieves the last name of the main collector with reference to the recordedBy field (in which case the CollectorDictionary field will be flagged as ‘checked’), otherwise, the function will return the last name of the main collector, extracted automatically from the recordedBy field .

Once verified, the collector’s dictionary can be reused in the future.

file.collectorsDictionary.dataset <- 'file\_collectorsDictionary\_dataset.csv'  
  
 write.csv(collectorsDictionary.dataset,  
 file.collectorsDictionary.dataset,   
 row.names = FALSE,   
 fileEncoding = "UTF-8",   
 na = "")

#### 3.3 Generating the collection event key

This generates a key to identify the physical and digital duplicates, of a given collection event. It combines the primary collector’s surname, the collector’s number and the botanical family, a key is created (family + recordByStandardized + recordNumber\_Standard) that allows grouping the duplicates of the same unique collection event.

It also identifiesnew collectors to be added to the collector dictionary and that can be reused in the future. in the future.

occ\_collectorsDictionary <- generate\_collection\_event\_key(occ=occ,  
 collectorDictionary\_checked\_file = file.collectorsDictionary.dataset)

## [1] "Loading collectorDictionary..."

names(occ\_collectorsDictionary)

## [1] "occ\_collectorsDictionary" "summary"   
## [3] "collectorsDictionary\_add"

head(occ\_collectorsDictionary$occ\_collectorsDictionary[,c(1,3)])

## # A tibble: 6 × 2  
## Ctrl\_nameRecordedBy\_Standard Ctrl\_key\_family\_recordedBy\_recordNumber  
## <chr> <chr>   
## 1 FONNEGRA ACHATOCARPACEAE\_FONNEGRA\_1629   
## 2 ROLDAN ACHATOCARPACEAE\_ROLDAN\_957   
## 3 FONNEGRA ACHATOCARPACEAE\_FONNEGRA\_1657   
## 4 BUNCH ACHATOCARPACEAE\_BUNCH\_   
## 5 DUQUE ACHATOCARPACEAE\_DUQUE\_4101   
## 6 TRUJILLO ACHATOCARPACEAE\_TRUJILLO\_5470

### 4. Selecting the master digital voucher

To group duplicates and choose the digital voucher:

Unique collection events can result in many ‘duplicate’ GBIF records. We designate one of these ‘duplicate’ records as the master digital voucher, to which data from other duplicate vouchers can be merged (see export\_data):

* **Where the collection event key for grouping duplicates is complete**, then duplicates can be grouped / parsed. To do so, we evaluate record completeness. Record completeness is calculated based on data-quality scores for the information in the following fields: recordedBy, recordNumber, year, institutionCode, catalogNumber, locality, municipality, countryCode, stateProvince and fieldNotes. The spatial coordinates associated with each duplicate are ranked using a score for the quality of the geospatial information. This score is calculated using the issues listed in the GBIF table, EnumOccurrenceIssue.  
  A score is calculated based on these issues (see above). The duplicate with the highest total score is assigned as the master voucher for the unique collection event. Missing information contained in duplicate records of the unique collection event can then be merged into the master digital voucher (see export\_data).
* **Where the collection event key is incomplete**, unique collection event duplicates cannot be parsed. In this case, each record is considered as a unique collection event, without duplicates. However, to know the integrity of the information, record completeness and quality of the geospatial information, are evaluated as described above.

**How is the quality score calculated?** parseGBIF\_digital\_voucher = The duplicate with the highest total score, sum of record completeness + quality of geospatial information.

**How is record completeness calculated?** The quality of the duplicate records associated with each collection event key is measured as the completeness of a record, using the sum of a number of flags (see below) equal to TRUE.

**Flags used to calculate record completeness**

* Is there information about the collector?
* Is there information about the collection number?
* Is there information about the year of collection?
* Is there information about the institution code?
* Is there information about the catalog number?
* Is there information about the locality?
* Is there information about the municipality of collection?
* Is there information about the state/province of collection?
* Is there information about the field notes?

**The quality of geospatial information is based on geographic issues raised by GBIF.** GIBF issues relating to geospatial data were classified into three classes based on the data quality scores that we assigned to each of the following GBIF issues recorded in the EnumOccurrenceIssue.

* Issue does not affect coordinating accuracy, with selection\_score equal to -1
* Issue has potential to affect coordinate accuracy, with selection\_score equal to -3
* Records with a selection\_score equal to -9 are excluded.

occ\_digital\_voucher <- parseGBIF::select\_digital\_voucher(occ = occ,  
 occ\_gbif\_issue = occ\_gbif\_issue$occ\_gbif\_issue,  
 occ\_wcvp\_check\_name = names.checked$occ\_wcvp\_check\_name ,  
 occ\_collectorsDictionary = occ\_collectorsDictionary$occ\_collectorsDictionary)  
  
 names(occ\_digital\_voucher)

## [1] "occ\_digital\_voucher" "occ\_results"

colnames(occ\_digital\_voucher$occ\_digital\_voucher)

## [1] "COORDINATE\_UNCERTAINTY\_METERS\_INVALID"   
## [2] "CONTINENT\_COORDINATE\_MISMATCH"   
## [3] "CONTINENT\_COUNTRY\_MISMATCH"   
## [4] "CONTINENT\_DERIVED\_FROM\_COORDINATES"   
## [5] "CONTINENT\_DERIVED\_FROM\_COUNTRY"   
## [6] "COORDINATE\_ACCURACY\_INVALID"   
## [7] "COORDINATE\_PRECISION\_INVALID"   
## [8] "COORDINATE\_PRECISION\_UNCERTAINTY\_MISMATCH"   
## [9] "COORDINATE\_REPROJECTED"   
## [10] "ELEVATION\_NON\_NUMERIC"   
## [11] "ELEVATION\_NOT\_METRIC"   
## [12] "ELEVATION\_UNLIKELY"   
## [13] "CONTINENT\_INVALID"   
## [14] "COUNTRY\_DERIVED\_FROM\_COORDINATES"   
## [15] "COUNTRY\_INVALID"   
## [16] "ELEVATION\_MIN\_MAX\_SWAPPED"   
## [17] "COORDINATE\_ROUNDED"   
## [18] "DEPTH\_MIN\_MAX\_SWAPPED"   
## [19] "DEPTH\_NON\_NUMERIC"   
## [20] "DEPTH\_NOT\_METRIC"   
## [21] "DEPTH\_UNLIKELY"   
## [22] "COUNTRY\_MISMATCH"   
## [23] "COORDINATE\_REPROJECTION\_FAILED"   
## [24] "COORDINATE\_REPROJECTION\_SUSPICIOUS"   
## [25] "GEODETIC\_DATUM\_INVALID"   
## [26] "PRESUMED\_NEGATED\_LATITUDE"   
## [27] "PRESUMED\_NEGATED\_LONGITUDE"   
## [28] "PRESUMED\_SWAPPED\_COORDINATE"   
## [29] "GEODETIC\_DATUM\_ASSUMED\_WGS84"   
## [30] "COORDINATE\_INVALID"   
## [31] "COORDINATE\_OUT\_OF\_RANGE"   
## [32] "COUNTRY\_COORDINATE\_MISMATCH"   
## [33] "ZERO\_COORDINATE"   
## [34] "AMBIGUOUS\_COLLECTION"   
## [35] "AMBIGUOUS\_INSTITUTION"   
## [36] "BASIS\_OF\_RECORD\_INVALID"   
## [37] "COLLECTION\_MATCH\_FUZZY"   
## [38] "COLLECTION\_MATCH\_NONE"   
## [39] "DIFFERENT\_OWNER\_INSTITUTION"   
## [40] "FOOTPRINT\_SRS\_INVALID"   
## [41] "FOOTPRINT\_WKT\_INVALID"   
## [42] "FOOTPRINT\_WKT\_MISMATCH"   
## [43] "GEOREFERENCED\_DATE\_INVALID"   
## [44] "GEOREFERENCED\_DATE\_UNLIKELY"   
## [45] "IDENTIFIED\_DATE\_INVALID"   
## [46] "IDENTIFIED\_DATE\_UNLIKELY"   
## [47] "INDIVIDUAL\_COUNT\_CONFLICTS\_WITH\_OCCURRENCE\_STATUS"  
## [48] "INDIVIDUAL\_COUNT\_INVALID"   
## [49] "INSTITUTION\_COLLECTION\_MISMATCH"   
## [50] "INSTITUTION\_MATCH\_FUZZY"   
## [51] "INSTITUTION\_MATCH\_NONE"   
## [52] "INTERPRETATION\_ERROR"   
## [53] "MODIFIED\_DATE\_INVALID"   
## [54] "MODIFIED\_DATE\_UNLIKELY"   
## [55] "MULTIMEDIA\_DATE\_INVALID"   
## [56] "MULTIMEDIA\_URI\_INVALID"   
## [57] "OCCURRENCE\_STATUS\_INFERRED\_FROM\_BASIS\_OF\_RECORD"   
## [58] "OCCURRENCE\_STATUS\_INFERRED\_FROM\_INDIVIDUAL\_COUNT"   
## [59] "OCCURRENCE\_STATUS\_UNPARSABLE"   
## [60] "POSSIBLY\_ON\_LOAN"   
## [61] "RECORDED\_DATE\_INVALID"   
## [62] "RECORDED\_DATE\_MISMATCH"   
## [63] "RECORDED\_DATE\_UNLIKELY"   
## [64] "REFERENCES\_URI\_INVALID"   
## [65] "TAXON\_MATCH\_AGGREGATE"   
## [66] "TAXON\_MATCH\_FUZZY"   
## [67] "TAXON\_MATCH\_HIGHERRANK"   
## [68] "TAXON\_MATCH\_NONE"   
## [69] "TYPE\_STATUS\_INVALID"   
## [70] "Ctrl\_gbifID"   
## [71] "Ctrl\_bibliographicCitation"   
## [72] "Ctrl\_language"   
## [73] "Ctrl\_institutionCode"   
## [74] "Ctrl\_collectionCode"   
## [75] "Ctrl\_datasetName"   
## [76] "Ctrl\_basisOfRecord"   
## [77] "Ctrl\_informationWithheld"   
## [78] "Ctrl\_dataGeneralizations"   
## [79] "Ctrl\_occurrenceID"   
## [80] "Ctrl\_catalogNumber"   
## [81] "Ctrl\_recordNumber"   
## [82] "Ctrl\_recordedBy"   
## [83] "Ctrl\_georeferenceVerificationStatus"   
## [84] "Ctrl\_occurrenceStatus"   
## [85] "Ctrl\_eventDate"   
## [86] "Ctrl\_year"   
## [87] "Ctrl\_month"   
## [88] "Ctrl\_day"   
## [89] "Ctrl\_habitat"   
## [90] "Ctrl\_fieldNotes"   
## [91] "Ctrl\_eventRemarks"   
## [92] "Ctrl\_locationID"   
## [93] "Ctrl\_higherGeography"   
## [94] "Ctrl\_islandGroup"   
## [95] "Ctrl\_island"   
## [96] "Ctrl\_countryCode"   
## [97] "Ctrl\_stateProvince"   
## [98] "Ctrl\_county"   
## [99] "Ctrl\_municipality"   
## [100] "Ctrl\_locality"   
## [101] "Ctrl\_verbatimLocality"   
## [102] "Ctrl\_locationRemarks"   
## [103] "Ctrl\_decimalLatitude"   
## [104] "Ctrl\_decimalLongitude"   
## [105] "Ctrl\_verbatimCoordinateSystem"   
## [106] "Ctrl\_verbatimIdentification"   
## [107] "Ctrl\_identificationQualifier"   
## [108] "Ctrl\_typeStatus"   
## [109] "Ctrl\_identifiedBy"   
## [110] "Ctrl\_dateIdentified"   
## [111] "Ctrl\_scientificName"   
## [112] "Ctrl\_family"   
## [113] "Ctrl\_taxonRank"   
## [114] "Ctrl\_nomenclaturalCode"   
## [115] "Ctrl\_taxonomicStatus"   
## [116] "Ctrl\_issue"   
## [117] "Ctrl\_mediaType"   
## [118] "Ctrl\_hasCoordinate"   
## [119] "Ctrl\_hasGeospatialIssues"   
## [120] "Ctrl\_verbatimScientificName"   
## [121] "Ctrl\_level0Name"   
## [122] "Ctrl\_level1Name"   
## [123] "Ctrl\_level2Name"   
## [124] "Ctrl\_level3Name"   
## [125] "wcvp\_plant\_name\_id"   
## [126] "wcvp\_taxon\_rank"   
## [127] "wcvp\_taxon\_status"   
## [128] "wcvp\_family"   
## [129] "wcvp\_taxon\_name"   
## [130] "wcvp\_taxon\_authors"   
## [131] "wcvp\_accepted\_plant\_name\_id"   
## [132] "wcvp\_reviewed"   
## [133] "wcvp\_searchedName"   
## [134] "wcvp\_taxon\_status\_of\_searchedName"   
## [135] "wcvp\_plant\_name\_id\_of\_searchedName"   
## [136] "wcvp\_taxon\_authors\_of\_searchedName"   
## [137] "wcvp\_verified\_author"   
## [138] "wcvp\_verified\_speciesName"   
## [139] "wcvp\_searchNotes"   
## [140] "Ctrl\_nameRecordedBy\_Standard"   
## [141] "Ctrl\_recordNumber\_Standard"   
## [142] "Ctrl\_key\_family\_recordedBy\_recordNumber"   
## [143] "Ctrl\_key\_year\_recordedBy\_recordNumber"   
## [144] "Ctrl\_geospatial\_quality"   
## [145] "Ctrl\_verbatim\_quality"   
## [146] "Ctrl\_moreInformativeRecord"   
## [147] "parseGBIF\_digital\_voucher"   
## [148] "parseGBIF\_duplicates"   
## [149] "parseGBIF\_num\_duplicates"   
## [150] "parseGBIF\_non\_groupable\_duplicates"   
## [151] "parseGBIF\_duplicates\_grouping\_status"   
## [152] "Ctrl\_coordinates\_validated\_by\_gbif\_issue"

file.occ\_digital\_voucher <- 'occ\_digital\_voucher.csv'  
  
 write.csv(occ\_digital\_voucher$occ\_digital\_voucher,  
 file.occ\_digital\_voucher,   
 row.names = FALSE,   
 fileEncoding = "UTF-8",   
 na = "")

### 5. Export of results

For each unique collection event key, complete or incomplete, outputs will be created which combine information from duplicate records and generate a single unique collection event record to replace them.

The main output fields relating to taxonomic identification and geographic coordinates:

* parseGBIF\_sample\_taxon\_name = scientific name chosen as taxonomic identification for unique collection event
* parseGBIF\_number\_taxon\_names = number of scientific names found in duplicates of unique collection event parseGBIF\_sample\_taxon\_name\_status = status of choice of ‘identified’, ‘divergent identifications’, ‘unidentified’
* parseGBIF\_unidentified\_sample = if unique collection event has taxonomic identification
* parseGBIF\_decimalLatitude = latitude in decimal degrees
* parseGBIF\_decimalLongitude = longitude in decimal degrees
* parseGBIF\_useful\_for\_spatial\_analysis = whether the coordinates are useful for spatial analysis.

**How is the taxon binomial attributed to the unique collection event selected?**

* **Where the unique collection event key is complete**: The accepted TAXON\_NAME selected is that which is most frequently applied to the duplicate vouchers at or below the rank of species. Where two named are applied with equal frequency then a mechanical approach, using alphabetical order, is applied, the first listed TAXON\_NAME being chosen. Where there is no identification, at or below the rank of species, then the unique collection event, the unique collection event is indicated as unidentified.
* **Where the unique collection event key is incomplete**: Where the unique collection event key is incomplete, then each record is treated as a unique collection event. If there is no identification, at or below the rank of species, then the unique collection event is classified as unidentified.

**Geospatial information**

If the master voucher does not have geographic coordinates, we will seek coordinates from the duplicate records associated with it. Finally, the records are separated into three sets of data:

* **useable\_data** Where unique collection event with taxonomic identification and geographic coordinates are complete. This represents the useable dataset.
* **unusable\_data** Where unique collection event without taxonomic identification and/or geographic coordinates.
* **duplicates** The duplicates of unique collection events complete / incomplete.

With this, it is possible to perform:

Merge information between fields of duplicates of a unique collection event to create a synthetic record for each unique collection event, Compare the frequency of content in fields Generate a work package summary.

For each complete unique collection event key, data fields that are empty in the digital voucher record will be populated with data from the respective duplicates. During content merging, we indicate fields associated with the description, location, and data of the unique collection event. By default, fields\_to\_merge parameter of export\_data function contains: - Ctrl\_fieldNotes - Ctrl\_year - Ctrl\_stateProvince - Ctrl\_municipality - Ctrl\_locality - Ctrl\_countryCode - Ctrl\_eventDate - Ctrl\_habitat - Ctrl\_level0Name - Ctrl\_level1Name - Ctrl\_level2Name - Ctrl\_level3Name

**export\_data function return a list with 10 data frames**:

* **all\_data** All records processed, merged Unique collection events complete / incomplete and their duplicates
* **useable\_data\_merge** Merged useable dataset
* **useable\_data\_raw** Raw useable dataset
* **duplicates** Duplicates of unique collection events of useable and useable datasets
* **unusable\_data\_merge** Merged unusable dataset. It is NA if merge\_unusable\_data is FALSE.
* **unusable\_data\_raw** Raw unusable dataset
* **parseGBIF\_general\_summary**
* **parseGBIF\_merge\_fields\_summary**
* **parseGBIF\_merge\_fields\_summary\_useable\_data**
* **parseGBIF\_merge\_fields\_summary\_unusable\_data** It is NA if merge\_unusable\_data is FALSE

results <- export\_data(occ\_digital\_voucher\_file = file.occ\_digital\_voucher,  
 merge\_unusable\_data = TRUE)  
  
names(results)

## [1] "all\_data"   
## [2] "useable\_data\_merge"   
## [3] "useable\_data\_raw"   
## [4] "duplicates"   
## [5] "unusable\_data\_merge"   
## [6] "unusable\_data\_raw"   
## [7] "parseGBIF\_general\_summary"   
## [8] "parseGBIF\_merge\_fields\_summary"   
## [9] "parseGBIF\_merge\_fields\_summary\_useable\_data"   
## [10] "parseGBIF\_merge\_fields\_summary\_unusable\_data"

results$parseGBIF\_general\_summary

## question value  
## 1 total number of records 3626  
## 2 total number of unique collection events 2777  
## 3 total number of unique collection events complete 1527  
## 4 total number of unique collection events incomplete 1250  
## 5 total number of duplicates 849  
## 6 total unique collection events containing merged fields 86

results$parseGBIF\_merge\_fields\_summary

## question value  
## 1 Ctrl\_habitat : total merge actions 41  
## 2 Ctrl\_locality : total merge actions 22  
## 3 Ctrl\_municipality : total merge actions 15  
## 4 Ctrl\_stateProvince : total merge actions 13  
## 5 Ctrl\_fieldNotes : total merge actions 12  
## 6 Ctrl\_year : total merge actions 11  
## 7 Ctrl\_eventDate : total merge actions 11  
## 8 Ctrl\_level0Name : total merge actions 2  
## 9 Ctrl\_level1Name : total merge actions 2  
## 10 Ctrl\_level2Name : total merge actions 2  
## 11 Ctrl\_level3Name : total merge actions 1  
## 12 Ctrl\_countryCode : total merge actions 0

results$parseGBIF\_merge\_fields\_summary\_complete

## NULL

NROW(results$all\_data)

## [1] 3626

NROW(results$unique\_collection\_event\_complete\_merge)

## [1] 0

NROW(results$unique\_collection\_event\_incomplete\_raw)

## [1] 0

NROW(results$duplicates)

## [1] 849

file.all <- 'parseGBIF\_all\_data.csv'  
write.csv(results$all\_data,  
 file.all,   
 row.names = FALSE,   
 fileEncoding = "UTF-8",   
 na = "")  
  
file.summary <- 'parseGBIF\_general\_summary.csv'  
write.csv(results$parseGBIF\_general\_summary,  
 file.summary,   
 row.names = FALSE,   
 fileEncoding = "UTF-8",   
 na = "")  
   
file.summary <- 'parseGBIF\_merge\_fields\_summary.csv'  
write.csv(results$parseGBIF\_merge\_fields\_summary,  
 file.summary,   
 row.names = FALSE,   
 fileEncoding = "UTF-8",   
 na = "")  
  
file.summary <- 'parseGBIF\_merge\_fields\_summary\_complete.csv'  
write.csv(results$parseGBIF\_merge\_fields\_summary\_complete,  
 file.summary,   
 row.names = FALSE,   
 fileEncoding = "UTF-8",   
 na = "")

### Accessing content frequency in fields and information about merged fields

* Merge information between fields of duplicates of a unique collection event

index <- results$useable\_data\_merge$Ctrl\_key\_family\_recordedBy\_recordNumber %in% 'ACHATOCARPACEAE\_ZARDINI\_5592'  
  
print('merged fields')

## [1] "merged fields"

print(jsonlite::fromJSON(results$useable\_data\_merge$parseGBIF\_duplicates\_map[index==TRUE]))

## $Ctrl\_gbifID  
## [1] "2996695631" "4061323610;[4061323610]"  
## [3] "1258412909;[1258412909]" "1095491559;[1095491559]"  
##   
## $Ctrl\_scientificName  
## [1] "Achatocarpus praecox Griseb."   
## [2] "Achatocarpus praecox var. bicornutus (Schinz & Autran) Botta;[4061323610]"  
## [3] "Achatocarpus praecox var. bicornutus (Schinz & Autran) Botta;[1258412909]"  
##   
## $Ctrl\_recordedBy  
## [1] "Zardini, EM;Zardini, E.M." "Elsa M. Zardini;[4061323610]"  
##   
## $Ctrl\_recordNumber  
## [1] "5592" "Zardini 5592;[4061323610]"  
##   
## $Ctrl\_identifiedBy  
## [1] "Liesner, R." "Ronald Liesner (MO);[4061323610]"  
## [3] "Ronald Liesner (MO);[1258412909]"  
##   
## $Ctrl\_dateIdentified  
## [1] "1990-01-01" "1996-01-01;[4061323610]"  
##   
## $Ctrl\_institutionCode  
## [1] "MBM" "FCQ;[4061323610]" "MO;[1258412909]"   
##   
## $Ctrl\_collectionCode  
## [1] "MBM" "Tropicos;[4061323610]"  
##   
## $Ctrl\_datasetName  
## [1] ""   
## [2] "Tropicos;[4061323610]"   
## [3] "MBM - Herbário do Museu Botânico Municipal;[1095491559]"  
##   
## $Ctrl\_datasetName  
## [1] ""   
## [2] "Tropicos;[4061323610]"   
## [3] "MBM - Herbário do Museu Botânico Municipal;[1095491559]"  
##   
## $Ctrl\_language  
## [1] "" "pt;[1095491559]"  
##   
## $wcvp\_plant\_name\_id  
## [1] "500159" "500160;[4061323610]"  
##   
## $wcvp\_taxon\_rank  
## [1] "Species" "Variety;[4061323610]"  
##   
## $wcvp\_taxon\_name  
## [1] "Achatocarpus praecox"   
## [2] "Achatocarpus praecox var. bicornutus;[4061323610]"  
##   
## $wcvp\_taxon\_authors  
## [1] "Griseb."   
## [2] "(Schinz & Autran) Botta;[4061323610]"  
## [3] "(Schinz & Autran) Botta;[1258412909]"  
##   
## $Ctrl\_stateProvince  
## [1] "Paraguari" "Paraguarí;[4061323610]"  
##   
## $Ctrl\_locality  
## [1] ""   
## [2] "Cerro Palacios. In forest.;[4061323610]"

print('merged fields map')

## [1] "merged fields map"

print(jsonlite::fromJSON(results$useable\_data\_merge$parseGBIF\_merged\_fields[index==TRUE]))

## $Ctrl\_locality  
## [1] "4061323610"

* Frequency of content in fields

print('Frequency of content in fields')

## [1] "Frequency of content in fields"

print(jsonlite::fromJSON(results$useable\_data\_merge$parseGBIF\_freq\_duplicate\_or\_missing\_data[index==TRUE]))

## $Ctrl\_gbifID  
## value freq  
## 1 1095491559 1  
## 2 1258412909 1  
## 3 2996695631 1  
## 4 4061323610 1  
##   
## $Ctrl\_scientificName  
## value freq  
## 1 Achatocarpus praecox Griseb. 2  
## 2 Achatocarpus praecox var. bicornutus (Schinz & Autran) Botta 2  
##   
## $Ctrl\_recordedBy  
## value freq  
## 1 Elsa M. Zardini 2  
## 2 Zardini, EM 1  
## 3 Zardini, EM;Zardini, E.M. 1  
##   
## $Ctrl\_recordNumber  
## value freq  
## 1 5592 2  
## 2 Zardini 5592 2  
##   
## $Ctrl\_identifiedBy  
## value freq  
## 1 Ronald Liesner (MO) 2  
## 2 Liesner, R 1  
## 3 Liesner, R. 1  
##   
## $Ctrl\_dateIdentified  
## value freq  
## 1 1990-01-01 2  
## 2 1996-01-01 2  
##   
## $Ctrl\_institutionCode  
## value freq  
## 1 MBM 2  
## 2 FCQ 1  
## 3 MO 1  
##   
## $Ctrl\_collectionCode  
## value freq  
## 1 MBM 2  
## 2 Tropicos 2  
##   
## $Ctrl\_datasetName  
## value freq  
## 1 Tropicos 2  
## 2 MBM - Herbário do Museu Botânico Municipal 1  
## 3 empty 1  
##   
## $Ctrl\_datasetName  
## value freq  
## 1 Tropicos 2  
## 2 MBM - Herbário do Museu Botânico Municipal 1  
## 3 empty 1  
##   
## $Ctrl\_language  
## value freq  
## 1 pt 1  
## 2 empty 3  
##   
## $wcvp\_plant\_name\_id  
## value freq  
## 1 500159 2  
## 2 500160 2  
##   
## $wcvp\_taxon\_rank  
## value freq  
## 1 Species 2  
## 2 Variety 2  
##   
## $wcvp\_taxon\_status  
## value freq  
## 1 Accepted 4  
##   
## $wcvp\_family  
## value freq  
## 1 Achatocarpaceae 4  
##   
## $wcvp\_taxon\_name  
## value freq  
## 1 Achatocarpus praecox 2  
## 2 Achatocarpus praecox var. bicornutus 2  
##   
## $wcvp\_taxon\_authors  
## value freq  
## 1 (Schinz & Autran) Botta 2  
## 2 Griseb. 2  
##   
## $wcvp\_reviewed  
## value freq  
## 1 Y 4  
##   
## $wcvp\_searchNotes  
## value freq  
## 1 Accepted 4  
##   
## $Ctrl\_fieldNotes  
## value freq  
## 1 Tree 8m.; Forest. aLT.: 250m. 1  
## 2 empty 3  
##   
## $Ctrl\_year  
## value freq  
## 1 1988 4  
##   
## $Ctrl\_stateProvince  
## value freq  
## 1 Paraguari 2  
## 2 Paraguarí 2  
##   
## $Ctrl\_municipality  
## value freq  
## 1 Cerro Palacios 2  
## 2 empty 2  
##   
## $Ctrl\_locality  
## value freq  
## 1 Cerro Palacios. In forest. 2  
## 2 empty 2  
##   
## $Ctrl\_countryCode  
## value freq  
## 1 PY 4  
##   
## $Ctrl\_eventDate  
## value freq  
## 1 1988-07-09 4  
##   
## $Ctrl\_habitat  
## value freq  
## 1 empty 4  
##   
## $Ctrl\_level0Name  
## value freq  
## 1 Paraguay 4  
##   
## $Ctrl\_level1Name  
## value freq  
## 1 Paraguarí 4  
##   
## $Ctrl\_level2Name  
## value freq  
## 1 Pirayú 4  
##   
## $Ctrl\_level3Name  
## value freq  
## 1 empty 4

1. Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais, [pablopains@yahoo.com.br](mailto:pablopains@yahoo.com.br) [↑](#footnote-ref-20)
2. Natural History Museum, London, [n\_bystriakova@yahoo.com](mailto:n_bystriakova@yahoo.com) [↑](#footnote-ref-22)
3. Royal Botanic Gardens, Kew, [a.monro@kew.org](mailto:a.monro@kew.org) [↑](#footnote-ref-24)