

# Package ‘ParsGBIF’

June 25, 2023

**Type** Package

**Title** An R package for parsing species occurrence records

**Version** 0.0.1

**Date** 2023-06-04

**Maintainer** Pablo Melo <pablopains@yahoo.com.br>

**Description** ParsGBIF package is designed to convert GBIF species occurrence data to a more comprehensible format to be used for further analysis, e.g. spatial.

The package provides tools for verifying and standardizing species scientific names and for selecting the most informative species records when duplicates are available.

**License** GPL (>= 2) | file LICENSE

**Encoding** UTF-8

**LazyData** true

**LazyDataCompression** xz

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**Imports** plyr,  
readxl,  
dplyr,  
tidyr,  
readr,  
stringr,  
textclean,  
googledrive,  
rvest,  
lubridate,  
rnatualearthdata,  
jsonlite,  
sqldf,  
DT,  
downloader,  
tidyselect,  
utils

**Remotes** github::pablopains/ParsGBIF

**Depends** R (>= 3.5.0)

## R topics documented:

batch_checkName_wcvp . . . . .	2
checkName_wcvp . . . . .	4
EnumOccurrenceIssue . . . . .	6
export_data . . . . .	7
extract_gbif_issue . . . . .	8
get_lastNameRecordedBy . . . . .	9
get_wcvp . . . . .	10
prepare_collectorsDictionary . . . . .	11
prepare_gbif_occurrence_data . . . . .	13
select_digital_voucher_and_sample_identification . . . . .	14
select_gbif_fields . . . . .	16
standardize_scientificName . . . . .	23
update_collectorsDictionary . . . . .	24
<b>Index</b>	<b>26</b>

---

batch_checkName_wcvp	<i>In batch, use the WCVp database to check accepted names and update synonyms</i>
----------------------	--

---

## Description

In batch, use the **World Checklist of Vascular Plants database (about WCVp)** to check accepted names and update synonyms

## Usage

```
batch_checkName_wcvp(
  occ = NA,
  wcvp_names = "",
  if_author_fails_try_without_combinations = TRUE,
  wcvp_selected_fields = "standard"
)
```

## Arguments

occ	GBIF occurrence table with selected columns as select_gbif_fields(columns = 'standard')
wcvp_names	get data frame in ParsGBIF::get_wcvp(read_only_to_memory = TRUE)\$wcvp_names or configure function to save a copy on local disk to optimize loading, see details in help(get_wcvp)
if_author_fails_try_without_combinations	option for partial verification of the authorship of the species. Remove the authors of combinations, in parentheses.
wcvp_selected_fields	WCVp fields selected as return, 'standard' basic columns, 'all' all available columns. The default is 'standard'

**Details**

See `help(checkName_wcvp)` and [about WCV database](#)

**Value**

list with two data frames: summary, species list and `occ_checkName_wcvp`, with WCV fields

**Author(s)**

Pablo Hendrigo Alves de Melo, Nadia Bystriakova & Alexandre Monro

**See Also**

[get\\_wcvp](#), [checkName\\_wcvp](#)

**Examples**

```
# These examples take >10 minutes to run and require 'ParsGBIF::get_wcvp()'

library(ParsGBIF)

help(batch_checkName_wcvp)

occ_file <- 'https://raw.githubusercontent.com/pablopains/ParsGBIF/main/dataGBIF/Achatocarpaceae/occurrence'

occ <- prepare_gbif_occurrence_data(gbif_occurrence_file = occ_file,
                                     columns = 'standard')

# wcvp_names <- get_wcvp(read_only_to_memory = TRUE)$wcvp_names
data(wcvp_names_Achatocarpaceae)

head(wcvp_names)

res_batch_checkName_wcvp <- batch_checkName_wcvp(occ = occ,
                                                wcvp_names = wcvp_names,
                                                if_author_fails_try_without_combinations = TRUE,
                                                wcvp_selected_fields = 'standard',
                                                show_process = TRUE)

names(res_batch_checkName_wcvp)

head(res_batch_checkName_wcvp$summary)

head(res_batch_checkName_wcvp$occ_checkName_wcvp)
```

---

checkName_wcvp	<i>Use the World Checklist of Vascular Plants (WCVP) database to check accepted names and update synonyms</i>
----------------	---

---

### Description

Use the **World Checklist of Vascular Plants database (about WCVP)** to check accepted names and update synonyms.

### Usage

```
checkName_wcvp(
  searchedName = "Hemistylus brasiliensis Wedd.",
  wcvp_names = "",
  if_author_fails_try_without_combinations = TRUE
)
```

### Arguments

searchedName	scientific name, with or without author
wcvp_names	WCVP table, wcvp_names.csv file from <a href="http://sftp.kew.org/pub/data-repositories/WCVP/">http://sftp.kew.org/pub/data-repositories/WCVP/</a> If NA, automatically load the latest version of the database by the function <code>ParsGBIF::get_wcvp(read_only_to_memory = TRUE)\$wcvp_names</code> .
if_author_fails_try_without_combinations	option for partial verification of the authorship of the species. Remove the authors of combinations, in parentheses

### Details

About the World Checklist of Vascular Plants <https://powo.science.kew.org/about-wcvp> search-Notes values:

- Accepted - When only one authorless scientific name is present in the list of TAXON\_name with and TAXON\_STATUS equal to "Accepted", verified\_speciesName = 100.
- Accepted among homonyms - When more than one authorless scientific name is present in the TAXON\_name list, but only one of the homonyms displays TAXON\_STATUS equal to "Accepted", verified\_speciesName = number of matches/100.
- Homonyms - When more than one authorless scientific name is present in the TAXON\_name list and more than one, or none among the homonyms, display TAXON\_STATUS equal to "Accepted", verified\_speciesName = number of matches/100. Before searching for homonyms, there was a failure in trying to find the matching match between authorless scientific name in TAXON\_name and author in TAXON\_AUTHORS, in these cases verified\_author equal to 0 (zero),
- Not Found: When the authorless scientific name is not present in the TAXON\_NAME LIST
- Unplaced: o When only one authorless scientific name is present in the list of TAXON\_name with and TAXON\_STATUS = "Unplaced"
- Updated: When only one authorless scientific name is present in the list of TAXON\_name and ACCEPTED\_PLANT\_NAME\_ID are not empty (and ACCEPTED\_PLANT\_NAME\_ID is different from the ID of the species consulted) taxon\_status\_of\_searchedName, plant\_name\_id\_of\_searchedName and taxon\_authors\_of\_searchedName values:

- When searchNotes equals "Updated" – The fields record the information of the scientific name originally consulted.
- When searchNotes equals "Homonyms" - Fields record the information of homonymous synonyms separated by "|".
- verified\_author values:
  - When value equal to 100 – when there is matched match between authorless scientific name in TAXON\_name and author in TAXON\_AUTHORS.
  - When value equal to 50 – when there is combined correspondence between authorless scientific name in TAXON\_name and author, without (combination), in TAXON\_AUTHORS.
  - When value equal to 0 – regardless of the correspondence between authorless scientific name in TAXON\_name, author is not present in TAXON\_AUTHORS.

## Value

Data frame with WCVF fields

## Author(s)

Pablo Hendrigo Alves de Melo, Nadia Bystriakova & Alexandre Monro

## See Also

[get\\_wcvp, standardize\\_scientificName](#)

## Examples

```
# These examples take >10 seconds to run and require 'ParsGBIF::get_wcvp()'
```

```
# load package
library(ParsGBIF)
```

```
help(checkName_wcvp)
```

```
wcvp_names <- get_wcvp(read_only_to_memory = TRUE)$wcvp_names
```

```
# 1) Updated
checkName_wcvp(searchedName = 'Hemistylus brasiliensis Wedd.',
  wcvp_names = wcvp_names,
  if_author_fails_try_without_combinations = TRUE)
```

```
# 2) Accepted
checkName_wcvp(searchedName = 'Hemistylus boehmerioides Wedd. ex Warm.',
  wcvp_names = wcvp_names,
  if_author_fails_try_without_combinations = TRUE)
```

```
# 3) Unplaced - taxon_status = Unplaced
checkName_wcvp(searchedName = 'Leucosyke australis Unruh',
  wcvp_names = wcvp_names,
  if_author_fails_try_without_combinations = TRUE)
```

```
# 4) Accepted among homonyms - When author is not informed. In this case, one of the homonyms, taxon_status is ac
checkName_wcvp(searchedName = 'Parietaria cretica',
  wcvp_names = wcvp_names,
  if_author_fails_try_without_combinations = TRUE)
```

```

# When author is informed
checkName_wcvp(searchedName = 'Parietaria cretica L.',
                wcvp_names = wcvp_names,
                if_author_fails_try_without_combinations = TRUE)

# When author is informed
checkName_wcvp(searchedName = 'Parietaria cretica Moris',
                wcvp_names = wcvp_names,
                if_author_fails_try_without_combinations = TRUE)

# 5) Homonyms - When author is not informed. In this case, none of the homonyms, taxon_status is Accepted
checkName_wcvp(searchedName = 'Laportea peltata',
                wcvp_names = wcvp_names,
                if_author_fails_try_without_combinations = TRUE)

# When author is informed
checkName_wcvp(searchedName = 'Laportea peltata Gaudich. & Decne.',
                wcvp_names = wcvp_names,
                if_author_fails_try_without_combinations = TRUE)

# When author is informed
checkName_wcvp(searchedName = 'Laportea peltata (Blume) Gaudich.',
                wcvp_names = wcvp_names,
                if_author_fails_try_without_combinations = TRUE)

```

---

EnumOccurrenceIssue	<i>Enumeration GBIF issue An enumeration of validation rules for single occurrence records.</i>
---------------------	---

---

## Description

There are many things that can go wrong and we continuously encounter unexpected data. In order to help us and publishers improve the data, we flag records with various issues that we have encountered. This is also very useful for data consumers as you can include these issues as filters in occurrence searches. Not all issues indicate bad data. Some are merely flagging the fact that GBIF has altered values during processing. On the details page of any occurrence record you will see the list of issues in the notice at the bottom.

## Usage

```
data(EnumOccurrenceIssue)
```

## Format

A data frame with 69 rows and 9 columns

## Details

**constant** GBIF issue constant

**description** GBIF issue description

**definition** Our definition for classifying geographic issues

**type** Type issue

**priority** Impact of the issue for the use of geospatial information

**score** Impact, in number, of the issue for the use of geospatial information

**selection\_score** Value used to calculate the quality of the geospatial information according to the classification of the issue

**reasoning** Reasoning of the impact of the theme for the use of geospatial information

**notes** Notes

## Source

- [GBIF Infrastructure: Data processing](#)
- [An enumeration of validation rules for single occurrence records](#)

---

export_data	<i>Export of results</i>
-------------	--------------------------

---

## Description

Separate records into three data frames Export of results:

- Useful data for spatial and taxonomic analysis
- Data in need of revision of spatial information or without identification
- Duplicates of the previous two datasets

## Usage

```
export_data(occ_digital_voucher_file = "", occ_digital_voucher = NA)
```

## Arguments

occ\_digital\_voucher\_file

CSV file result of function select\_digital\_voucher\_and\_sample\_identification()\$occ\_digital\_voucher

occ\_digital\_voucher

data frame result of function select\_digital\_voucher\_and\_sample\_identification()\$occ\_digital\_voucher

## Details

Each data frame should be used as needed

## Value

list with three data frames: occ\_in, Useful data for spatial and taxonomic analysis, occ\_out\_to\_recover, data in need of spatial data revision or without identification and occ\_dup, duplicates.

## Author(s)

Pablo Hendrigo Alves de Melo, Nadia Bystrakova & Alexandre Monro

## See Also

[batch\\_checkName\\_wcvp](#), [extract\\_gbif\\_issue](#)





```
data(EnumOccurrenceIssue)

colnames(EnumOccurrenceIssue)

head(EnumOccurrenceIssue)

occ_gbif_issue <- extract_gbif_issue(occ = occ)

names(occ_gbif_issue)

head(occ_gbif_issue$summary)

colnames(occ_gbif_issue$occ_gbif_issue)

head(occ_gbif_issue$occ_gbif_issue)
```

---

```
get_lastNameRecordedBy
      get_lastNameRecordedBy
```

---

**Description**

Returns the last name of the main collector

**Usage**

```
get_lastNameRecordedBy(x)
```

**Arguments**

x                      recordedBy field

**Details**

Returns the last name of the main collector in recordedBy field

**Value**

last name of the main collector

**Author(s)**

Pablo Hendrigo Alves de Melo, Nadia Bystriakova & Alexandre Monro

**See Also**

[prepare\\_collectorsDictionary](#), [update\\_collectorsDictionary](#)

**Examples**

```
help(get_lastNameRecordedBy)

get_lastNameRecordedBy('Melo, P.H.A & Monroe, A.')

get_lastNameRecordedBy('Monroe, A. & Melo, P.H.A')
```

get\_wcvp

*Get WCVP database***Description**

Download World Checklist of Vascular Plants (WCVP) database

**Usage**

```
get_wcvp(
  url_source = "http://sftp.kew.org/pub/data-repositories/WCVP/",
  read_only_to_memory = FALSE,
  path_results = "C:/ParsGBIF",
  update = FALSE,
  load_distribution = FALSE
)
```

**Arguments**

url_source	http://sftp.kew.org/pub/data-repositories/WCVP/
read_only_to_memory	TRUE to in-memory read-only, not writing a copy to local disk
path_results	download destination folder, if read_only_to_memory FALSE
update	TRUE to update and load files, FALSE to keep local version and load files, if read_only_to_memory FALSE
load_distribution	TRUE to load file with geographical distribution of species, if read_only_to_memory FALSE

**Details**

<http://sftp.kew.org/pub/data-repositories/WCVP/> This is the public SFTP (Secure File Transfer Protocol) site of the Royal Botanic Gardens, Kew. This space contains data resources publicly accessible to the user 'anonymous'. No password required for access. Use of data made available via this site may be subject to legal and licensing restrictions. The README in the top-level directory for each data resource provides specific information about its terms of use.

**Value**

list with two data frames: wcvp\_names and wcvp\_distribution

**Author(s)**

Pablo Hendrigo Alves de Melo, Nadia Bystriakova & Alexandre Monro

**See Also**

[checkName\\_wcvp](#), [standardize\\_scientificName](#)

**Examples**

```
# load package
library(ParsGBIF)

help(get_wcvp)

# Download wcvp database to local disk
path_data <- tempdir() # you can change this folder

wcvp <- get_wcvp(url_source = 'http://sftp.kew.org/pub/data-repositories/WCVP/',
                 read_only_to_memory = FALSE,
                 path_results = path_data,
                 update = FALSE,
                 load_distribution = TRUE)

names(wcvp)

head(wcvp$wcvp_names)
colnames(wcvp$wcvp_names)

head(wcvp$wcvp_distribution)
colnames(wcvp$wcvp_distribution)
```

---

```
prepare_collectorsDictionary
      prepare_collectorsDictionary
```

---

**Description**

Returns the list with the last name of the main collector associated with the unique key recordedBy.

**Usage**

```
prepare_collectorsDictionary(
  occ = NA,
  collectorDictionary_file =
    "https://raw.githubusercontent.com/pablopains/ParsGBIF/main/collectorDictionary/CollectorsDic
  )
```

**Arguments**

`occ` GBIF occurrence table with selected columns as `select_gbif_fields(columns = 'standard')`

`collectorDictionary_file` Collector dictionary file - point to a file on your local disk or upload via git at <https://raw.githubusercontent.com/pablopains/ParsGBIF/main/collectorDictionary/CollectorsDictionary>

**Details**

If `recordedBy` is present in the collector's dictionary, it returns the checked name, if not, it returns the last name of the main collector, extracted from the `recordedBy` field. If `recordedBy` is present in the collector's dictionary, returns the main collector's last name associated with the single `recordedBy` key, otherwise, returns the main collector's last name, extracted from the `recordedBy` field. It is recommended to curate the main collector's surname, automatically extracted from the `recordedBy` field. The objective is to standardize the last name of the main collector. That the primary botanical collector of a sample is always recognized by the same last name, standardized in capital letters and non-ascii characters replaced

**Value**

`Ctrl_nameRecordedBy_Standard`, `Ctrl_recordedBy`, `Ctrl_notes`, `collectorDictionary`, `Ctrl_update`, `collectorName`, `Ctrl_fullName`, `Ctrl_fullNameII`, `CVStarrVirtualHerbarium_PersonDetails`

**Author(s)**

Pablo Hendrigo Alves de Melo, Nadia Bystriakova & Alexandre Monro

**See Also**

[select\\_gbif\\_fields](#), [update\\_collectorsDictionary](#)

**Examples**

```
help(prepare_collectorsDictionary)

occ <- prepare_gbif_occurrence_data(gbif_occurrence_file = 'https://raw.githubusercontent.com/pablopains/ParsGBIF/main/occurrence/occurrence.csv',
                                   columns = 'standard')

collectorsDictionaryFromDataset <- prepare_collectorsDictionary(occ = occ,
                                                                collectorDictionary_file = 'https://raw.githubusercontent.com/pablopains/ParsGBIF/main/collectorDictionary/CollectorsDictionary.csv')

colnames(collectorsDictionaryFromDataset)
head(collectorsDictionaryFromDataset)

collectorDictionary_checked_file <- paste0(tempdir(), '/', 'collectorsDictionaryFromDataset.csv')

collectorDictionary_checked_file

write.csv(collectorsDictionaryFromDataset,
          collectorDictionary_checked_file,
          row.names = FALSE,
          fileEncoding = "UTF-8",
          na = "")
```

---

```
prepare_gbif_occurrence_data
```

*Prepare occurrence data from GBIF to use in package*

---

## Description

Prepare occurrence data downloaded from GBIF to be used by ParsGBIF functions

## Usage

```
prepare_gbif_occurrence_data(gbif_occurrence_file = "", columns = "standard")
```

## Arguments

`gbif_occurrence_file`

The name of the file from which the with occurrence data downloaded from GBIF (by default "occurrence.txt")

`columns`

Character vector of strings to indicate column names of the GBIF occurrence file. Use 'standard' to select basic columns for use in the package, 'all' to select all available columns. The default is 'standard'

## Details

Select data fields and rename field names prefixed with "Ctrl\_"

## Value

data.frame with renamed selected fields with prefix "Ctrl\_"

## Author(s)

Pablo Hendrigo Alves de Melo, Nadia Bystriakova & Alexandre Monro

## See Also

[select\\_gbif\\_fields](#), [extract\\_gbif\\_issue](#)

## Examples

```
library(ParsGBIF)
```

```
help(prepare_gbif_occurrence_data)
```

```
occ_file <- 'https://raw.githubusercontent.com/pablopains/ParsGBIF/main/dataGBIF/Achatocarpaceae/occurrence'
```

```
occ <- prepare_gbif_occurrence_data(gbif_occurrence_file = occ_file,
                                   columns = 'standard')
```

```
colnames(occ)
```

```
head(occ)
```

---

select\_digital\_voucher\_and\_sample\_identification

*Select a sample among available duplicates*

---

## Description

To group duplicates and choose the digital voucher:

1. If the key for grouping duplicates is complete with collector information and collection number, sample duplicates can be grouped. In this case, the voucher with the highest score is selected among the duplicates in the sample.
2. If the key to group duplicates is incomplete, sample duplicates cannot be grouped due to missing collector information and/or collection number. In this case, each record is considered a sample, without duplicates, and a voucher is selected for each sample.

How is the information score calculated?

moreInformativeRecord = sum of textual quality + quality of geospatial information.

How is the quality of textual information calculated?

The Text quality is the sum of the number of flags with text quality equal to TRUE.

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 collection site? Is there information about the municipality of collection? Is there information about the state/province of collection? Is there information about the bibliographic citation?

How is the quality of geospatial information calculated?

The quality of geospatial information is based on geographic issues made available by GBIF.

GBIF issues on the quality of geospatial information were classified into three levels.

- Not applicable, with selection\_score equal to 0
- Does not affect coordinating accuracy, with selection\_score equal to -1
- Potentially affect coordinate accuracy, with selection\_score equal to -3
- Records to be excluded from spatial analysis, with selection\_score equal to -9

How is the taxonomic identification of the sample chosen?

1. When the key to group the duplicates is complete: The accepted TAXON\_NAME identified at or below the specified level and the most frequent among the duplicates is chosen.

In case of a tie in frequency, in alphabetical order, the first accepted TAXON\_NAME identified up to or below the specific level is chosen.

If there is no identification, equal to or less than the specific level, for the sample, the sample is indicated as unidentified.

1. When the key to group the duplicates is incomplete: If so, the accepted TAXON\_NAME identified at or below the specified level is used. If there is no identification, equal to or less than the specific level, the sample is indicated as unidentified.

**Usage**

```
select_digital_voucher_and_sample_identification(
  occ = NA,
  occ_gbif_issue = NA,
  occ_checkName_wcvp = NA,
  occ_collectorsDictionary = NA,
  enumOccurrenceIssue = NA
)
```

**Arguments**

**occ** GBIF occurrence table with selected columns as `select_gbif_fields(columns = 'standard')`

**occ\_gbif\_issue** = result of function `extract_gbif_issue()``$occ_gbif_issue`

**occ\_checkName\_wcvp** = result of function `batch_checkName_wcvp()``$occ_checkName_wcvp`

**occ\_collectorsDictionary** = result of function `update_collectorsDictionary()``$occ_collectorsDictionary`

**enumOccurrenceIssue** An enumeration of validation rules for single occurrence records by GBIF file, if NA, will be used, `data(EnumOccurrenceIssue)`

**Details**

- **matchStatusDuplicates** - "matched", "unmatched: no recordedBy and no recordNumber", "unmatched: no recordNumber" or "unmatched: no recordedBy"
- **numberTaxonNamesSample** - count of the different accepted scientific names, identified up to or below the specific level, listed in the sample duplicates, or Zero, if there is no identification, equal to or below the specific level, for the sample.
- **sampleTaxonName** - TAXON\_name accepted and identified up to or below the specific level selected for the sample.
- **sampleIdentificationStatus** - 'Identified', 'divergent identifications', or 'unidentified'

**Value**

list with two data frames: `occ_digital_voucher_and`: `occ_digital_voucher_and_sample_identification`, only with selection result fields and `occ_join_results`, with all data processing fields.

**Author(s)**

Pablo Hendrigo Alves de Melo, Nadia Bystriakova & Alexandre Monro

**See Also**

[batch\\_checkName\\_wcvp](#), [extract\\_gbif\\_issue](#)

**Examples**

```
help(select_digital_voucher_and_sample_identification)

head(occ)
head(res_gbif_issue$occ_gbif_issue)
```

```
head(res_checkName_wcvp$occ_checkName_wcvp)
head(res_collectorsDictionary$occ_collectorsDictionary)
res_digital_voucher_and_sample_identification <- select_digital_voucher_and_sample_identification(occ = occ,
                                                    occ_gbif_issue = res_gbif_issue$occ_gbif_i
                                                    occ_checkName_wcvp = res_checkName_wcvp$occ
                                                    occ_collectorsDictionary = res_collectorsD
                                                    enumOccurrenceIssue = EnumOccurrenceIssue)

names(res_digital_voucher_and_sample_identification)

head(res_digital_voucher_and_sample_identification$occ_digital_voucher_and_sample_identification)
colnames(res_digital_voucher_and_sample_identification$occ_digital_voucher_and_sample_identification)

head(res_digital_voucher_and_sample_identification$occ_join_results)
colnames(res_digital_voucher_and_sample_identification$occ_join_results)
```

---

select_gbif_fields	<i>select_gbif_fields</i>
--------------------	---------------------------

---

**Description**

Select columns in GBIF occurrence data

**Usage**

```
select_gbif_fields(columns = "standard")
```

**Arguments**

columns	'standard' basic columns about what, when, where, and who collected, 'all' all available columns or list column names
---------	---

**Details**

"standard" : indicated by (**standard**)  
or  
'all':

- 'gbifID'
- 'abstract'
- 'accessRights'
- 'accrualMethod'
- 'accrualPeriodicity'
- 'accrualPolicy'
- 'alternative'
- 'audience'
- 'available'
- 'bibliographicCitation' (**standard**)



- 'conformsTo'
- 'contributor'
- 'coverage'
- 'created'
- 'creator'
- 'date'
- 'dateAccepted'
- 'dateCopyrighted'
- 'dateSubmitted'
- 'description'
- 'educationLevel'
- 'extent'
- 'format'
- 'hasFormat'
- 'hasPart'
- 'hasVersion'
- 'identifier'
- 'instructionalMethod'
- 'isFormatOf'
- 'isPartOf'
- 'isReferencedBy'
- 'isReplacedBy'
- 'isRequiredBy'
- 'isVersionOf'
- 'issued'
- 'language' (**standard**)
- 'license'
- 'mediator'
- 'medium'
- 'modified'
- 'provenance'
- 'publisher'
- 'references'
- 'relation'
- 'replaces'
- 'requires'
- 'rights'
- 'rightsHolder'
- 'source'
- 'spatial'

- 'subject'
- 'tableOfContents'
- 'temporal'
- 'title'
- 'type'
- 'valid'
- 'institutionID'
- 'collectionID'
- 'datasetID'
- 'institutionCode' (**standard**)
- 'collectionCode' (**standard**)
- 'datasetName' (**standard**)
- 'ownerInstitutionCode'
- 'basisOfRecord' (**standard**)
- 'informationWithheld' (**standard**)
- 'dataGeneralizations' (**standard**)
- 'dynamicProperties'
- 'occurrenceID' (**standard**) # `occ_search(occurrenceId='BRA:UNEMAT:HPAN:6089')`
- 'catalogNumber' (**standard**)
- 'recordNumber' (**standard**)
- 'recordedBy' (**standard**)
- 'recordedByID'
- 'individualCount'
- 'organismQuantity'
- 'organismQuantityType'
- 'sex'
- 'lifeStage'
- 'reproductiveCondition'
- 'behavior'
- 'establishmentMeans'
- 'degreeOfEstablishment'
- 'pathway'
- 'georeferenceVerificationStatus' (**standard**)
- 'occurrenceStatus' (**standard**)
- 'preparations'
- 'disposition'
- 'associatedOccurrences'
- 'associatedReferences'
- 'associatedSequences'
- 'associatedTaxa'

- 'otherCatalogNumbers'
- 'occurrenceRemarks'
- 'organismID'
- 'organismName'
- 'organismScope'
- 'associatedOrganisms'
- 'previousIdentifications'
- 'organismRemarks'
- 'materialSampleID'
- 'eventID'
- 'parentEventID'
- 'fieldNumber'
- 'eventDate' (**standard**)
- 'eventTime'
- 'startDayOfYear'
- 'endDayOfYear'
- 'year' (**standard**)
- 'month' (**standard**)
- 'day' (**standard**)
- 'verbatimEventDate'
- 'habitat' (**standard**)
- 'samplingProtocol'
- 'sampleSizeValue'
- 'sampleSizeUnit'
- 'samplingEffort'
- 'fieldNotes' (**standard**)
- 'eventRemarks' (**standard**)
- 'locationID' (**standard**)
- 'higherGeographyID'
- 'higherGeography' (**standard**)
- 'continent'
- 'waterBody'
- 'islandGroup' (**standard**)
- 'island' (**standard**)
- 'countryCode' (**standard**)
- 'stateProvince' (**standard**)
- 'county' (**standard**)
- 'municipality' (**standard**)
- 'locality' (**standard**)
- 'verbatimLocality' (**standard**)

- 'verbatimElevation'
- 'verticalDatum'
- 'verbatimDepth'
- 'minimumDistanceAboveSurfaceInMeters'
- 'maximumDistanceAboveSurfaceInMeters'
- 'locationAccordingTo'
- 'locationRemarks' (**standard**)
- 'decimalLatitude' (**standard**)
- 'decimalLongitude' (**standard**)
- 'coordinateUncertaintyInMeters'
- 'coordinatePrecision'
- 'pointRadiusSpatialFit'
- 'verbatimCoordinateSystem' (**standard**)
- 'verbatimSRS'
- 'footprintWKT'
- 'footprintSRS'
- 'footprintSpatialFit'
- 'georeferencedBy'
- 'georeferencedDate'
- 'georeferenceProtocol'
- 'georeferenceSources'
- 'georeferenceRemarks'
- 'geologicalContextID'
- 'earliestEonOrLowestEonothem'
- 'latestEonOrHighestEonothem'
- 'earliestEraOrLowestErathem'
- 'latestEraOrHighestErathem'
- 'earliestPeriodOrLowestSystem'
- 'latestPeriodOrHighestSystem'
- 'earliestEpochOrLowestSeries'
- 'latestEpochOrHighestSeries'
- 'earliestAgeOrLowestStage'
- 'latestAgeOrHighestStage'
- 'lowestBiostratigraphicZone'
- 'highestBiostratigraphicZone'
- 'lithostratigraphicTerms'
- 'group'
- 'formation'
- 'member'
- 'bed'

- 'identificationID'
- 'verbatimIdentification' (**standard**)
- 'identificationQualifier' (**standard**)
- 'typeStatus' (**standard**)
- 'identifiedBy' (**standard**)
- 'identifiedByID'
- 'dateIdentified' (**standard**)
- 'identificationReferences'
- 'identificationVerificationStatus'
- 'identificationRemarks'
- 'taxonID'
- 'scientificNameID'
- 'acceptedNameUsageID'
- 'parentNameUsageID'
- 'originalNameUsageID'
- 'nameAccordingToID'
- 'namePublishedInID'
- 'taxonConceptID'
- 'scientificName' (**standard**)
- 'acceptedNameUsage'
- 'parentNameUsage'
- 'originalNameUsage'
- 'nameAccordingTo'
- 'namePublishedIn'
- 'namePublishedIn Year'
- 'higherClassification'
- 'kingdom'
- 'phylum'
- 'class'
- 'order'
- 'family' (**standard**)
- 'subfamily'
- 'genus'
- 'genericName'
- 'subgenus'
- 'infragenericEpithet'
- 'specificEpithet'
- 'infraspecificEpithet'
- 'cultivarEpithet'
- 'taxonRank' (**standard**)

- 'verbatimTaxonRank'
- 'vernacularName'
- 'nomenclaturalCode' (**standard**)
- 'taxonomicStatus' (**standard**)
- 'nomenclaturalStatus'
- 'taxonRemarks'
- 'datasetKey'
- 'publishingCountry'
- 'lastInterpreted'
- 'elevation'
- 'elevationAccuracy'
- 'depth'
- 'depthAccuracy'
- 'distanceAboveSurface'
- 'distanceAboveSurfaceAccuracy'
- 'issue' (**standard**)
- 'mediaType' (**standard**)
- 'hasCoordinate' (**standard**)
- 'hasGeospatialIssues' (**standard**)
- 'taxonKey'
- 'acceptedTaxonKey'
- 'kingdomKey'
- 'phylumKey'
- 'classKey'
- 'orderKey'
- 'familyKey'
- 'genusKey'
- 'subgenusKey'
- 'speciesKey'
- 'species'
- 'acceptedScientificName'
- 'verbatimScientificName' (**standard**)
- 'typifiedName'
- 'protocol'
- 'lastParsed'
- 'lastCrawled'
- 'repatriated'
- 'relativeOrganismQuantity'
- 'level0Gid'
- 'level0Name' (**standard**)

- 'level1Gid'
- 'level1Name' (**standard**)
- 'level2Gid'
- 'level2Name' (**standard**)
- 'level3Gid'
- 'level3Name' (**standard**)
- 'iucnRedListCategory'

**Value**

list of the columns names

**Author(s)**

Pablo Hendrigo Alves de Melo, Nadia Bystriakova & Alexandre Monro

**See Also**

[extract\\_gbif\\_issue](#), [prepare\\_gbif\\_occurrence\\_data](#)

**Examples**

```
# select_gbif_fields()

help(select_gbif_fields)

col_sel <- select_gbif_fields(columns = 'all')

col_sel <- select_gbif_fields(columns = 'standard')
```

---

```
standardize_scientificName
      standardize_scientificName
```

---

**Description**

standardize binomial name, variety, subspecies, form and hybrids, authorship to allow comparison with names of taxa in the World Checklist of Vascular Plants (WCVF) database

**Usage**

```
standardize_scientificName(
  searchedName = "Alomia angustata (Gardner) Benth. ex Baker"
)
```

**Arguments**

searchedName      scientific name, with or without author

**Details**

Standardize scientific name according to WCVF format. Separate generic epithet, specific epithet, variety, subspecies, form, hybrid and author, in the scientific name, if any. Standardize, according to WCVF, abbreviation of infrataxon, if any: variety to var., subspecies to subsp., FORM to f., hybrid separator separate x from the specific epithet.

**Value**

searchedName, standardizedName, taxonAuthors, taxonAuthors\_last

**Author(s)**

Pablo Hendrigo Alves de Melo, Nadia Bystriakova & Alexandre Monro

**See Also**

[get\\_wcvf](#), [checkName\\_wcvf](#)

**Examples**

```
# standardize_scientificName()

help(standardize_scientificName)

standardize_scientificName('Leucanthemum xsuperbum (Bergmans ex J.W.Ingram) D.H.Kent')
standardize_scientificName('Alomia angustata (Gardner) Benth. ex Baker')
standardize_scientificName('Centaurea xaemiliae Font Quer')
```

---

update\_collectorsDictionary

*Create a key to group duplicates of a sample*

---

**Description**

Include recordedByStandardized field with verified main collector's last name. Include recordNumber\_Standard field with only numbers from recordNumber. Create a key to group duplicates in the key\_family\_recordedBy\_recordNumber field, composed of the fields: family + recordedByStandardized + recordNumber\_Standard.

**Usage**

```
update_collectorsDictionary(
  occ = NA,
  collectorDictionary_checked_file = NA,
  collectorDictionary_file =
    "https://raw.githubusercontent.com/pablopains/ParsGBIF/main/collectorDictionary/CollectorsDic
)
```



**Arguments**

`occ` GBIF occurrence table with selected columns as `select_gbif_fields(columns = 'standard')`

`collectorDictionary_checked_file` Verified collector dictionary file - point to a file on your local disk

`collectorDictionary_file` Collector dictionary file - point to a file on your local disk or upload via git at <https://raw.githubusercontent.com/pablopains/ParsGBIF/main/collectorDictionary/CollectorsDictionary>

**Details**

Fields created for each incident record: `nameRecordedBy_Standard`, `recordNumber_Standard`, `key_family_recordedBy_Standard`, `key_year_recordedBy_recordNumber`

**Value**

list with three data frames: `occ_collectorsDictionary`, with update result fields only, summary and `CollectorsDictionary_add`, with new collectors that can be added to the collector dictionary that can be reused in the future.

**Author(s)**

Pablo Hendrigo Alves de Melo, Nadia Bystriakova & Alexandre Monro

**See Also**

[select\\_gbif\\_fields](#), [prepare\\_collectorsDictionary](#)

**Examples**

```
collectorsDictionaryFromDataset <- prepare_lastNameRecordedBy(occ=occ,  
  collectorDictionary_checked_file='collectorDictionary_checked.csv')
```

# Index

## \* datasets

EnumOccurrenceIssue, [6](#)

batch\_checkName\_wcvp, [2](#), [7](#), [15](#)

checkName\_wcvp, [3](#), [4](#), [11](#), [24](#)

EnumOccurrenceIssue, [6](#)

export\_data, [7](#)

extract\_gbif\_issue, [7](#), [8](#), [13](#), [15](#), [23](#)

get\_lastNameRecordedBy, [9](#)

get\_wcvp, [3](#), [5](#), [10](#), [24](#)

prepare\_collectorsDictionary, [9](#), [11](#), [25](#)

prepare\_gbif\_occurrence\_data, [8](#), [13](#), [23](#)

select\_digital\_voucher\_and\_sample\_identification,  
[14](#)

select\_gbif\_fields, [8](#), [12](#), [13](#), [16](#), [25](#)

standardize\_scientificName, [5](#), [11](#), [23](#)

update\_collectorsDictionary, [9](#), [12](#), [24](#)