

# Package ‘ParsGBIF’

June 15, 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 . . . . .	3
EnumOccurrenceIssue . . . . .	6
extract_gbif_issue . . . . .	6
get_lastNameRecordedBy . . . . .	8
get_wcvp . . . . .	9
prepare_collectorsDictionary . . . . .	10
prepare_gbif_occurrence_data . . . . .	11
select_digital_voucher_and_sample_identification . . . . .	13
select_gbif_fields . . . . .	14
standardize_scientificName . . . . .	16
update_collectorsDictionary . . . . .	17

<b>Index</b>	<b>19</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 WCVp database](#)

**Value**

list with two data frames: summary, species list and occ\_checkName\_wcvp, with WCVp 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

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

---

**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 <http://sftp.kew.org/pub/data-repositories/WCVp/>  
If NA, automatically load the latest version of the database by the function `ParsGBIF::get_wcvp(read_only_to_memory = TRUE)$wcvp_names`.

**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 WCVp 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</i>
---------------------	-------------------------------

---

## Description

An enumeration of validation rules for single occurrence records.

## Usage

```
EnumOccurrenceIssue
```

## Format

EnumOccurrenceIssue:

A data frame with 69 rows and 9 columns:

**Constant** GBIF Constant

**Description** Description

**definition** definition

**type** type

**priority** priority

**score** score

**selection\_score** selection\_score

**reasoning** reasoning

**notes** notes

## Source

<https://gbif.github.io/parsers/apidocs/org/gbif/api/vocabulary/OccurrenceIssue.html>

---

extract_gbif_issue	<i>Extract GBIF issue occurrence records</i>
--------------------	--

---

## Description

Extract GBIF validation rules for occurrence records

## Usage

```
extract_gbif_issue(occ = NA, enumOccurrenceIssue = NA)
```

**Arguments**

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

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

**Details**

<https://gbif.github.io/parsers/apidocs/org/gbif/api/vocabulary/OccurrenceIssue.html>

**Value**

list with two data frames: `summary`, with the frequency of issues in the records and `occ_gbif_issue`, with issues in columns with TRUE or FALSE for each record.

**Author(s)**

Pablo Hendrigo Alves de Melo, Nadia Bystriakova & Alexandre Monro

**See Also**

[prepare\\_gbif\\_occurrence\\_data](#), [select\\_gbif\\_fields](#)

**Examples**

```
library(ParsGBIF)

help(extract_gbif_issue)

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

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

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 & Monro, A.')

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



---

`get_wcvp``get_wcvp`

---

### 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

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

README\_WCVp.xlsx, wcvp\_distribution.csv, wcvp\_names.csv

### 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**

<code>occ</code>	GBIF occurrence table with selected columns as <code>select_gbif_fields(columns = 'standard')</code>
<code>collectorDictionary_file</code>	Collector dictionary file - point to a file on your local disk or upload via git at <a href="https://raw.githubusercontent.com/pablopains/ParsGBIF/main/collectorDictionary/CollectorsDictionary">https://raw.githubusercontent.com/pablopains/ParsGBIF/main/collectorDictionary/CollectorsDictionary</a>

**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/Par
                                columns = 'standard')

collectorsDictionaryFromDataset <- prepare_collectorsDictionary(occ = occ,
                                collectorDictionary_file = 'https://raw.githubusercontent.com/pa

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 GBIF herbarium specimen occurrence data for use in the package

**Usage**

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

**Arguments**

<code>gbif_occurrence_file</code>	the name of the file from which the GBIF herbarium specimen occurrence data is to be read
<code>columns</code>	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_digital_voucher_and_sample_identification
```

---

## Description

Extract gbif issue

## 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

To group duplicates: 1) If the key to group duplicates is incomplete: Sample duplicates cannot be grouped due to missing collector information and/or collection number. Each record is considered a sample, with no duplicates. Select a voucher for each sample. Or, 2) If the key to group duplicates is complete: Group duplicates. Select a voucher, with the highest information score, among the duplicates in the sample. How to calculate the information score?  $\text{moreInformativeRecord} = \text{sum of verbatim quality} + \text{quality of geospatial information}$ .  $\text{verbatim quality} = \text{sum of the number of flags with verbatim quality equal to TRUE}$ .  $\text{quality of geospatial information} = \text{If there is a geospatial issue, consider the one with the highest priority, with the highest score}$ . To select sample taxonomic identification: 1) If the key for grouping duplicates is complete: select the accepted TAXON\_NAME, identified up to or below the specific level, most frequent among the duplicates in the sample. If tie between frequency of accepted TAXON\_NAME, identified up to or below the specific level: select the first accepted TAXON\_NAME, identified up to or below the specific level, in alphabetical order. If there is no identification, at or below the specific level, for the sample: Indicate as unidentified sample. Or, 2) If the key for grouping duplicates is incomplete: select TAXON\_NAME, if accepted and identified up to or below the specified level. If there is no identification, at or below the specific level, for the sample: Indicate as unidentified sample.

**Value**

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'

**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$oc
                                                    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: c('bibliographicCitation', 'language', 'institutionCode', 'collectionCode', 'datasetName', 'basisOfRecord', 'informationWithheld', 'dataGeneralizations', 'occurrenceID', '# occ\_search(occurrenceId=BRA:UN', 'catalogNumber', 'recordNumber', 'recordedBy', 'georeferenceVerificationStatus', 'occurrenceStatus', 'eventDate', 'year', 'month', 'day', 'habitat', 'fieldNotes', 'eventRemarks', 'locationID', 'higherGeography', 'islandGroup', 'island', 'countryCode', 'stateProvince', 'county', 'municipality', 'locality', 'verbatimLocality', 'locationRemarks', 'decimalLatitude', 'decimalLongitude', 'verbatimCoordinateSystem', 'verbatimIdentification', 'identificationQualifier', 'typeStatus', 'identifiedBy', 'dateIdentified', 'scientificName', 'family', 'taxonRank', 'nomenclaturalCode', 'taxonomicStatus', 'issue', 'mediaType', 'hasCoordinate', 'hasGeospatialIssues', 'verbatimScientificName', 'level0Name', 'level1Name', 'level2Name', 'level3Name')

'all': c('gbifID', 'abstract', 'accessRights', 'accrualMethod', 'accrualPeriodicity', 'accrualPolicy', 'alternative', 'audience', 'available', 'bibliographicCitation', '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', 'license', 'mediator', 'medium', 'modified', 'provenance', 'publisher', 'references', 'relation', 'replaces', 'requires', 'rights', 'rightsHolder', 'source', 'spatial', 'subject', 'tableOfContents', 'temporal', 'title', 'type', 'valid', 'institutionID', 'collectionID', 'datasetID', 'institutionCode', 'collectionCode', 'datasetName', 'ownerInstitutionCode', 'basisOfRecord', 'informationWithheld', 'dataGeneralizations', 'dynamicProperties', 'occurrenceID', 'catalogNumber', 'recordNumber', 'recordedBy', 'recordedByID', 'individualCount', 'organismQuantity', 'organismQuantityType', 'sex', 'lifeStage', 'reproductiveCondition', 'behavior', 'establishmentMeans', 'degreeOfEstablishment', 'pathway', 'georeferenceVerificationStatus', 'occurrenceStatus', 'preparations', 'disposition', 'associatedOccurrences', 'associatedReferences', 'associatedSequences', 'associatedTaxa', 'otherCatalogNumbers', 'occurrenceRemarks', 'organismID', 'organismName', 'organismScope', 'associatedOrganisms', 'previousIdentifications', 'organismRemarks', 'materialSampleID', 'eventID', 'parentEventID', 'fieldNumber', 'eventDate', 'eventTime', 'startDayOfYear', 'endDayOfYear', 'year', 'month', 'day', 'verbatimEventDate', 'habitat', 'samplingProtocol', 'sampleSizeValue', 'sampleSizeUnit', 'samplingEffort', 'fieldNotes', 'eventRemarks', 'locationID', 'higherGeographyID', 'higherGeography', 'continent', 'waterBody', 'islandGroup', 'island', 'countryCode', 'stateProvince', 'county', 'municipality', 'locality', 'verbatimLocality', 'verbatimElevation', 'verticalDatum', 'verbatimDepth', 'minimumDistanceAboveSurfaceInMeters', 'maximumDistanceAboveSurfaceInMeters', 'locationAccordingTo', 'locationRemarks', 'decimalLatitude', 'decimalLongitude', 'coordinateUncertaintyInMeters', 'coordinatePrecision', 'pointRadiusSpatialFit', 'verbatimCoordinateSystem', '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', 'identificationQualifier', 'typeStatus', 'identifiedBy', 'identifiedByID', 'dateIdentified', 'identificationReferences', 'identificationVerificationStatus', 'identificationRemarks', 'taxonID', 'scientificNameID', 'acceptedNameUsageID', 'parentNameUsageID', 'originalNameUsageID', 'nameAccordingToID', 'namePublishedInID', 'taxonConceptID', 'scientificName', 'acceptedNameUsage', 'parentNameUsage', 'originalNameUsage', 'nameAccordingTo', 'namePublishedIn', 'namePub-

lishedInYear', 'higherClassification', 'kingdom', 'phylum', 'class', 'order', 'family', 'subfamily', 'genus', 'genericName', 'subgenus', 'infragenericEpithet', 'specificEpithet', 'infraspecificEpithet', 'cultivarEpithet', 'taxonRank', 'verbatimTaxonRank', 'vernacularName', 'nomenclaturalCode', 'taxonomicStatus', 'nomenclaturalStatus', 'taxonRemarks', 'datasetKey', 'publishingCountry', 'lastInterpreted', 'elevation', 'elevationAccuracy', 'depth', 'depthAccuracy', 'distanceAboveSurface', 'distanceAboveSurfaceAccuracy', 'issue', 'mediaType', 'hasCoordinate', 'hasGeospatialIssues', 'taxonKey', 'acceptedTaxonKey', 'kingdomKey', 'phylumKey', 'classKey', 'orderKey', 'familyKey', 'genusKey', 'subgenusKey', 'speciesKey', 'species', 'acceptedScientificName', 'verbatimScientificName', 'typifiedName', 'protocol', 'lastParsed', 'lastCrawled', 'repatriated', 'relativeOrganismQuantity', 'level0Gid', 'level0Name', 'level1Gid', 'level1Name', 'level2Gid', 'level2Name', 'level3Gid', 'level3Name', '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, standardizeName, 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 *superbum (Bergmans ex J.W.Ingram) D.H.Kent')
standardize_scientificName('Alomia angustata (Gardner) Benth. ex Baker')
standardize_scientificName('Centaurea *aemiliae Font Quer')
```

---

update\_collectorsDictionary  
*update\_collectorsDictionary*

---

**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**

....

**Value**

`occ_collectorsDictionary`: `nameRecordedBy_Standard`, `recordNumber_Standard`, `key_family_recordedBy_recordNumber`, `key_year_recordedBy_recordNumber` summary, `collectorsDictionary_add`

**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](#), [14](#)

checkName\_wcvp, [3](#), [3](#), [9](#), [17](#)

EnumOccurrenceIssue, [6](#)

extract\_gbif\_issue, [6](#), [12](#), [14](#), [16](#)

get\_lastNameRecordedBy, [8](#)

get\_wcvp, [3](#), [5](#), [9](#), [17](#)

prepare\_collectorsDictionary, [8](#), [10](#), [18](#)

prepare\_gbif\_occurrence\_data, [7](#), [11](#), [16](#)

select\_digital\_voucher\_and\_sample\_identification,  
[13](#)

select\_gbif\_fields, [7](#), [11](#), [12](#), [14](#), [18](#)

standardize\_scientificName, [5](#), [9](#), [16](#)

update\_collectorsDictionary, [8](#), [11](#), [17](#)