package 'finbif'

January 16, 2020

Title Interface for the 'Finnish Biodiversity Information Facility' API

Version 0.2.0

Description A programmatic interface to the 'Finnish Biodiversity Information Facility' ('FinBIF') API (https://api.laji.fi). 'FinBIF' aggregates Finnish biodiversity data from multiple sources in a single open access portal for researchers, citizen scientists, industry and government. 'FinBIF' allows users of biodiversity information to find, access, combine and visualise data on Finnish plants, animals and microorganisms. The 'finbif' package makes the publicly available data in 'FinBIF' easily accessible to programmers. Biodiversity information is available on taxonomy and taxon occurrence. Occurrence data can be filtered by taxon, time, location and other variables. The data accessed are conveniently preformatted for subsequent analyses.

License MIT + file LICENSE


BugReports https://github.com/luomus/finbif/issues

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyData true

Depends R (>= 3.5.0)

Imports digest, httr, graphics, jsonlite, lubridate, lutz, methods, utils

RoxygenNote 7.0.2

Suggests codemetar, data.tree, details, ISOcodes, knitr, rmarkdown, ows4R, sf, stats, testthat (>= 2.1.0), usethis, vcr, vdiff

X-schema.org-applicationCategory Biodiversity

X-schema.org-keywords api, biodiversity, biodiversity-informatics, biodiversity-information, finbif, finbif-access, occurrences, r-package, r-programming, rstats, species, specimens, taxon, taxonomy, web-services
**as.data.frame.finbif_records**

*Coerce a finbif_records object to a data.frame*

**Description**

Converts the result of a FinBIF query to a data.frame.

**Usage**

```r
## S3 method for class 'finbif_records'
as.data.frame(x, ...)  

## S3 method for class 'finbif_records_list'
as.data.frame(x, ..., quiet = TRUE)
```
Arguments

x A `finbif_records` object.

quiet Logical. If `TRUE` (default) suppress progress indicator of conversion.

Value

A `data.frame`.

Examples

```r
## Not run:
# Download the latest records from FinBIF
# and convert to a `data.frame`
resp <- finbif_records()
df <- as.data.frame(resp)
## End(Not run)
```

---

**breaks_xy**

Create XY Breakpoints

Description

Creates sets of equally spaced breakpoints for XY coordinate data.

Usage

`breaks_xy(bbox, size)`

Arguments

bbox Numeric vector. A vector of points of the form, `c(xmin, ymin, xmax, ymax)` giving the outer limits of the breakpoints when expanded to nearest multiple of `size`.

size Numeric. The size of the cells between the breakpoints.

Value

A list with three components:

- x The X dimension breakpoints.
- y The Y dimension breakpoints.

Examples

```r
breaks_xy(c(5, -45, 67, 100), 10)
```
caching

Caching FinBIF downloads

Description

Working with cached data from FinBIF.

Turning caching off

By default, local filesystem caching of most FinBIF API requests is turned on. Any request made using the same arguments will only request data from FinBIF in the first instance and subsequent requests will use the local cache while it exists. This will increase the speed of repeated requests and save bandwidth and computation for the FinBIF server. Caching can be turned off temporarily by setting the `cache = FALSE` in the requesting function.

Setting options(`finbif_use_cache = FALSE`) will turn off caching for the current session.

Setting the cache directory

By default cached requests are stored in `tempdir()`. This can be changed by setting the file path for the current session with `options(finbif_cache_path = "path/to/cache")`.

Clearing the cache

The cache files can be removed using `finbif_clear_cache()`.

filters

Filtering FinBIF records

Description

Filters available for FinBIF records and occurrence data.

Taxa

Filters related to taxa include:

- `taxon_id` Character vector. FinBIF taxon IDs. The functions `finbif_check_taxa()` and `finbif_taxa()` can be used to search for taxon IDs.
- `taxon_name` Character vector. Filter based on taxon names (scientific or common) rather than IDs. If the specified taxa are not found in the FinBIF taxonomy then matches are attempted with the occurrence record names as originally supplied verbatim.
- `quality_controlled_det` Logical. If `TRUE` (default) use quality controlled taxonomic determinations. Or, if `FALSE` use the originally recorded taxonomic determinations.
• **subtaxa** Logical. If TRUE (default) return records of all taxa belonging to specified taxa. Or, if FALSE only return records for exact matches to the specified taxa (e.g., if a genus is specified, do not return records of the species belonging to the genus, return records of individuals identified as that genus only and not identified to a lower taxonomic level).

• **invalid_taxa** Logical. If TRUE (default) return records for taxa not found in the FinBIF taxonomic database as well as taxa that are in the FinBIF database. Or, if FALSE limit records to only those of taxa found in the FinBIF database.

• **informal_group** Character vector. Filter by informal taxonomic groups. Only including informal groups linked to the recorded taxa in the FinBIF database. Use the function `finbif_informal_groups()` to see the informal taxonomic groups available in FinBIF.

• **informal_group_reported** Character vector. Filter by informal taxonomic groups including groups reported directly with the record and those linked to the recorded taxa in the FinBIF database. Use the function `finbif_informal_groups()` to see the informal taxonomic groups available in FinBIF.

• **administrative_status** Character vector. Filter by administrative status code. Use the function `finbif_metadata()` to see administrative statuses and codes.

• **red_list_status** Character vector. Filter by IUCN red list status code. Use the function `finbif_metadata()` to see red list statuses and codes.

• **primary_habitat** Character vector or named list of character vectors. Filter by primary habitat code. Use the function `finbif_metadata()` to see habitat (sub)types and codes for taxa in the FinBIF database. Habitat type/subtypes can be refined further by indicating habitat qualifiers with a named list of character vectors where the names are habitat (sub)type codes and the elements of the character vector are the habitat qualifier codes. Use the function `finbif_metadata()` to see habitat qualifiers and codes. The records returned will be of taxa whose primary habitat is considered to be the (sub)habitat/habitat qualifier combination supplied.

• **primary_secondary_habitat** Character or named list of character vectors. As above, except the records returned will be of taxa whose primary or secondary habitat is considered to be the combination supplied.

• **finnish_occurrence_status** Character vector. Filter by Finnish occurrence status of taxa. Use `finbif_metadata()` to see the possible occurrence statuses of taxa.

• **finnish_occurrence_status_neg** Character vector. Negation of the above. Selecting a status will filter out rather than include records with the selected status.

• **finnish** Logical. If TRUE, limit records to taxa thought to occur in Finland. Or if FALSE limit to taxa not thought to occur in Finland. If unspecified (default) return records of all taxa.

• **invasive** Logical. If TRUE, limit records to invasive taxa. Or if FALSE limit to non-invasive taxa. If unspecified (default) return records of invasive and non-invasive taxa.

• **taxon_rank** Character vector. Filter by taxonomic rank. Use `finbif_metadata()` to see the taxonomic ranks available. Records returned will be limited to the specified ranks and not include records of lower taxonomic levels.

**Location**

Filters related to location of record include:
• **locality** Character vector. Filter by name of locality. Will first try to match strings to the countries, provinces, and municipalities (see below) in FinBIF, if none of these locality types match exactly then will return records with verbatim locality matches in the original records.

• **country** Character vector. Filter by country. Use `finbif_metadata()` to see country names and ISO codes (2 and 3 character) used in FinBIF.

• **province** Character vector. Filter by province. Use `finbif_metadata()` to see province names and codes.

• **municipality** Character vector. Filter by municipality. Use `finbif_metadata()` to see municipality names.

• **bird_assoc_area** Character vector. Filter by BirdLife Finland association area. Use `finbif_metadata()` to see association names and codes.

• **coordinates** Coordinates. A character vector or list of coordinate data. Must be length 3 to 4 (e.g., `list(lat = c(60.4,61),lon = c(22,22.5),system = "wgs84",ratio = 1)`). The first element is minimum and maximum latitude and the second minimum and maximum longitude (or can be minimums only). The third element is the coordinate system; either one of "wgs84", "euref" or "kkj". The optional fourth element is a positive value less than 1. When 1, the coverage area of the returned records will be completely within the box bound by the coordinates values. Values less than 1 requires the returned record’s coverage to overlap with the bounding box in that proportion. When using the system "kkj" the coordinates will be coerced to integers with units inferred from the number of integer digits (7 digits equals km’s, 6 equals 10km’s, etc.). If coordinate maximums are not specified they will be assumed to be one unit above the minimums (e.g., `c(666,333,"kkj")` is equivalent to `list(c(6660000,6670000),c(3330000,3340000),"kkj")`).

• **coordinates_center** Coordinates. A character vector or list of coordinate data. Must be of length 3. The first two elements are latitude and longitude and third is the coordinate system (currently only "wgs84" is implemented). Records returned will be those for which the center point exactly matches that which is specified.

• **coordinates_cell_{1k|10k|50k|100k}** Coordinates. A vector of coordinate data (lat, long). Filter by grid cell at scale * . Where * is 1, 10, 50 or 100. The coordinates specify the southeast corner of the cell. Coordinates system is uniform "kkj" (also known as "ykj").

• **coordinates_cell_{1k|10k|50k|100k}_center** Coordinates. As above, except coordinates indicate center of grid cell.

• **coordinates_source** Character. Filter by source of coordinates. Currently accepted values are "reported_value" (coordinates were recorded at time of observation) and "finnish_municipality" (coordinates were derived and observer only recorded municipality).

### Time

Filters related to time of record include:

• **date_range_ymd** Dates. An `Interval` object or a vector of one to two `Date` objects (begin and end dates) or objects that are coercible to the `Date` class by `as_date`. When supplying dates as strings, the day or month-and-day can be omitted (e.g., "2001-04" or "2001"). Note however, that when omitting day, only "-" is allowed to separate year and month, and months must be in two-digit/leading zero form. If the begin or end dates are partial date strings they will be interpreted as the first or last day of the month or year (e.g., `c(2001,2003)` is equivalent to `c("2001-01-01","2003-12-12")`). If a single date is supplied as a partial date string then all
records that fall within that month or year will be returned (e.g., c("2001-01") is equivalent to c("2001-01-01", "2001-01-31").

• date_range_ym Dates. As above, but days (if supplied) will be ignored.

• date_range_d Integer vector. Filter by day of the year (e.g., 1 to 366). If begin or end date is omitted then it is interpreted as the first or last day of the year.

• date_range_md Character vector. Filter by month and day of the year (e.g., "01-01" to "12-31"). If begin or end date is omitted then it is interpreted as the first or last day of the year.

• [first|last]_import_date_{min|max} Date. Filter by date record was imported. A Date object or object that is coercible to the Date class by as.Date.

Quality

Filters related to quality of record:

• collection_reliability Integer. Filter by collection reliability rating, from low reliability (1) to highly reliable (5).

• coordinates_uncertainty_max Integer. Filter by maximum uncertainty of coordinates (i.e., coordinates_uncertainty_max = 100 will return records that are accurate to 100m).

• quality_issues. Character. Filter by the presence of record quality issues. One of "without_issues", "with_issues" or "both". Issues include any quality issues with the record, the event, or the document. The default is "without_issues" unless filtering by record, event or document ID or record annotation status.

• reliable Logical. Filter by reliability of record. The "collection_reliability" is 4-5 or the record has been "confirmed" and has no quality issues (record, event or document).

• taxon_reliability Character vector. Filter by reliability of taxon identification. Can be one or more of "reliable", "likely", "neutral", "unlikely" or "unreliable".

Misc

Other filters:

• keywords Character vector. Filter by keywords.

• collection Character vector or finbif_collections() data.frame. Filter by collection. If a character vector can refer to collection ID, collection name (in English) or abbreviated name. Use finbif_collections() to see list of collections and metadata. Can also use the results of a call to finbif_collections() directly to filter records.

• subcollections Logical. If TRUE (default) include the subcollections of the collections specified. If FALSE do not include subcollections.

• not_collection Character vector or finbif_collections() data.frame. As for collection, but result will be the negation of the specified collections.

• source Character vector. Filter by information system data source. Use finbif_metadata() to see data source IDs names and descriptions.

• record_basis Character vector. Filter by basis of record. Use finbif_metadata() to see list of record bases.
• **superrecord_basis** Character vector. Filter by superset of record basis. One or more of "human_observation", "machine_observation", or "specimen".

• **life_stage** Character vector. Filter by organism life stage. Use `finbif_metadata()` to see list of organism life stages.

• **sex** Character vector. Filter by organism sex and sex-related category name or code. Use `finbif_metadata()` to see list of organism sexes and sex-related categories and codes. If "male" or "female" is specified then records returned will be those with sex specified as male or female respectively and those records where the corresponding {male/female}_abundance > 1.

• **event_id** Character. Filter by event (list of records, etc.) ID.

• **document_id** Character. Filter by the document (collection of events) ID of occurrences.

• **record_id** Character.

• **individual_id** Character. Filter by individual (an individual organism) ID.

• **abundance_min** Integer. Filter by the minimum number of individual organisms in the record.

• **abundance_max** Integer. Filter by the maximum number of individual organisms in the record.

• **type_specimen** Logical. Filter by whether or not the record is a type specimen.

• **wild_status** Character. Filter by "wildness" status of records. One or more of "wild", "non_wild" or "unknown". Default is c("wild","unknown").

• **is_breeding_location** Logical. Filter by whether or not the occurrence is recorded at a known breeding location.

• **has_document_media** Logical. Filter by whether there is media (images, video, audio, etc.) associated with the records’ document.

• **has_event_media** Logical. Filter by whether there is media (images, video, audio, etc.) associated with the records’ event.

• **has_record_media** Logical. Filter by whether there is media (images, video, audio, etc.) associated with the record.

• **has_media** Logical. Filter by whether there is any media (images, video, audio, etc.) associated with the record, its document or its event.

• **event_observer_name** Character. Filter by observer name.

• **event Observer_id** Integer. Filter by observer ID.

• **restriction_reason** Character vector. Filter by reason data has security restrictions. See `finbif_metadata()` for a list of reasons data may have security restrictions.

• **restriction_level** Character vector. Filter by data restriction level. See `finbif_metadata()` for a list of the levels of data restrictions.

• **restricted** Logical. Filter records by whether any data restrictions are in place (TRUE) or not (FALSE).

• **annotated** Logical. Filter records that do (TRUE) or do not (FALSE) have annotations.

• **unidentified** Logical. Filter by whether the record has been identified to species level and linked to the FinBIF taxon database (FALSE) or has not been identified to species level reliably and linked to the taxon database (TRUE).

• **taxon_census** Character vector. Return records belonging to surveys or censuses of a given taxon or taxonomic group. Specify the taxonomic group with a FinBIF taxon ID. Use `finbif_check_taxa()` to find taxon IDs.
finbif_check_taxa

• [record|event|document]_fact Character vector. Filter by record, event or document facts. Facts are key-value pairs of the form "<fact>=<value>". Value can be omitted in which case all records with any value recorded for the specified fact will be returned.
• has_sample Logical. Record includes a sample or samples (e.g., a DNA sample or preparation)

Description
Check that taxa are in the FinBIF database.

Usage
finbif_check_taxa(taxa, cache = getOption("finbif_use_cache"))

Arguments
taxa Character (or list of named character) vector(s). If a list each vector can have the name of a taxonomic rank (genus, species, etc.). The elements of the vectors should be the taxa to check.
cache Logical. Use cached data.

Value
An object of class finbif_taxa. A list with the same form as taxa.

Examples
## Not run:

# Check a scientific name
finbif_check_taxa("Cygnus cygnus")

# Check a common name
finbif_check_taxa("Whooper swan")

# Check a genus
finbif_check_taxa("Cygnus")

# Check a list of taxa
finbif_check_taxa(
  list(
    species = c("Cygnus cygnus", "Ursus arctos"),
    genus = "Betula"
  )
)

## End(Not run)
### finbif_clear_cache  Clear cache

**Description**
Remove cached API requests from FinBIF.

**Usage**

```r
finbif_clear_cache()
```

**Value**
0 for success, 1 for failure, invisibly.

**Examples**

```r
## Not run:
finbif_clear_cache()
## End(Not run)
```

### finbif_collections  FinBIF collections

**Description**
Get information on collections in the FinBIF database.

**Usage**

```r
finbif_collections(
  filter,
  select,
  subcollections = TRUE,
  supercollections = FALSE,
  lang = c("en", "fi", "sv"),
  nmin = 0,
  cache = getOption("finbif_use_cache")
)
```
finbif_informal_groups

Arguments

- **filter**
  Logical. Expression indicating elements or rows to keep: missing values are taken as false.

- **select**
  Expression. Indicates columns to select from the data frame.

- **subcollections**
  Logical. Return subcollection metadata of higher level collections.

- **supercollections**
  Logical. Return lowest level collection metadata.

- **lang**
  Character. Language of data returned. One of "en", "fi", or "sv".

- **nmin**
  Integer. Filter collections by number of records. Only return information on collections with greater than value specified. If NA then return information on all collections.

- **cache**
  Logical. Use cached data.

Value

A data.frame.

Examples

```r
## Not run:
# Get collection metadata
collections <- finbif_collections()
## End(Not run)
```

---

### finbif_informal_groups

#### FinBIF informal groups

**Description**

Display the informal taxonomic groups used in the FinBIF database.

**Usage**

```r
finbif_informal_groups(group, limit = 50, quiet = FALSE)
```

**Arguments**

- **group**
  Character. Optional, if supplied only display this group and its subgroups.

- **limit**
  Integer. The maximum number informal groups to display.

- **quiet**
  Logical. Return informal group names without displaying them.
finbif_metadata

Description
Display metadata from the FinBIF database.

Usage
finbif_metadata(which)

Arguments
which Character. Which category of metadata to display. If unspecified, function returns the categories of metadata available.

Value
A data.frame.

Examples
finbif_metadata("red_list")
finbif_occurrence

Download FinBIF occurrence records

Description

Download filtered occurrence data from FinBIF as a data.frame.

Usage

finbif_occurrence(
  ..., 
  filter, 
  select, 
  order_by, 
  sample = FALSE, 
  n = 10, 
  page = 1, 
  count_only = FALSE, 
  quiet = FALSE, 
  cache = getOption("finbif_use_cache"), 
  check_taxa = TRUE, 
  on_check_fail = c("warn", "error", "quiet"), 
  date_time = TRUE, 
  date_time_method = "fast", 
  tzone = Sys.timezone(), 
  dwc = FALSE 
)

Arguments

... Character vectors or list of character vectors. Taxa of records to download.
filter List of named character vectors. Filters to apply to records.
select Character vector. Variables to return. If not specified a default set of commonly used variables will be used. Use "default_vars" as a shortcut for this set.
order_by Character vector. Variables to order records by before they are returned. Most, though not all, variables can be used to order records before they are returned. Ordering is ascending by default. To return in descending order append a - to the front of the variable (e.g., "-date_start"). Default order is "-date_start" > "-load_data" > "reported_name".
sample Logical. If TRUE randomly sample the records from the FinBIF database.
n Integer. How many records to download.
page Integer. Which page of records to start downloading from.
count_only Logical. Only return the number of records available.
quiet Logical. Suppress the progress indicator for multipage downloads.
cache Logical. Use cached data.

check_taxa Logical. Check first that taxa are in the FinBIF database. If true only records that match known taxa (have a valid taxon ID) are returned.

on_check_fail Character. What to do if a taxon is found not valid. One of "warn" (default), "error" or "continue".

date_time Logical. Convert raw date and time variables into date-time and duration.

date_time_method Character. Passed to lutz::tz_lookup_coords() when date_time = TRUE. Default is "fast". Use date_time_method = "accurate" (requires package sf) for greater accuracy.

tzone Character. If date_time = TRUE the timezone of outputted date-time. Defaults to system timezone.

dwc Logical. Return Darwin Core (or Darwin Core style) variable names.

Value

A data.frame. If count_only = TRUE an integer.

Examples

## Not run:

# Get recent occurrence data for taxon
finbif_occurrence("Cygnus cygnus")

# Specify the number of records
finbif_occurrence("Cygnus cygnus", n = 100)

# Get multiple taxa
finbif_occurrence("Cygnus cygnus", "Ursus arctos")

# Filter the records
finbif_occurrence(
  species = "Cygnus cygnus",
  filter = list(coordinate_accuracy_max = 100)
)

## End(Not run)

---

**finbif_records**  
*Get FinBIF records*

**Description**

Download records from FinBIF.
Usage

finbif_records(
    filter,
    select,
    order_by,
    sample = FALSE,
    n = 10,
    page = 1,
    count_only = FALSE,
    quiet = FALSE,
    cache = getOption("finbif_use_cache")
)

Arguments

filter List of named character vectors. Filters to apply to records.
select Character vector. Variables to return. If not specified a default set of commonly
    used variables will be used. Use "default_vars" as a shortcut for this set.
order_by Character vector. Variables to order records by before they are returned. Most,
    though not all, variables can be used to order records before they are returned.
    Ordering is ascending by default. To return in descending order append a - to the
    front of the variable (e.g., "-date_start"). Default order is 
    
    n Integer. How many records to download.
page Integer. Which page of records to start downloading from.
count_only Logical. Only return the number of records available.
quiet Logical. Suppress the progress indicator for multipage downloads.
cache Logical. Use cached data.

Value

A finbif_api or finbif_api_list object.

Examples

## Not run:

# Get the last 100 records from FinBIF
finbif_records(n = 100)

## End(Not run)
finbif_request_token  
*Get a FinBIF personal access token*

**Description**

Have a personal access token for use with the FinBIF API sent to a specified email address.

**Usage**

```
finbif_request_token(email)
```

**Arguments**

- `email`  
  Character. The email address to which to send the API access token.

**Value**

If an access token has already been set then `NULL` (invisibly) if not then, invisibly, a `finbif_api` object containing the response from the FinBIF server.

**Examples**

```r
## Not run:

# Request a token for example@email.com
finbif_request_token("example@email.com")

## End(Not run)
```

---

finbif_taxa  
*Search the FinBIF taxa*

**Description**

Search the FinBIF database for taxon.

**Usage**

```
finbif_taxa(
  name,
  n = 1,
  type = c("exact", "partial", "likely"),
  cache = getOption("finbif_use_cache")
)
```
Arguments

name  Character. The name of a taxon to search for.
n  Integer. Maximum number of matches to return. For types "exact" and "likely" a maximum of one taxon will be returned.
type  Character. Type of match to make. Must be one of exact, partial or likely.
cache  Logical. Use cached data.

Value

A finbif_api object.

Examples

## Not run:

```r
# Search for a taxon
finbif_taxa("Ursus arctos")

# Use partial matching
finbif_taxa("Ursus", n = 10, "partial")
```

## End(Not run)

---

finland_map  Border of Finland

Description

Vertices of the Finnish border at the (approximately) 1km scale.

Usage

finland_map

Format

A list:

- vertices  A matrix of the vertices, in decimal degrees (WGS84) of the Finnish border at the (approximately) 1km scale.
- bbox  A vector of coordinates for a box bounding the Finnish border.

Source

https://www.stat.fi/org/avoindata/paikkatietoaineistot.html
Examples

library(finbif)
with(
  finland_map,
  (plot.new()
  plot.window(
    bbox[c(1, 3)],
    bbox[c(2, 4)],
    asp = 2.4
  )
  polygon(vertices))
)

hist_xy

2-Dimensional Histogram

Description

Compute a histogram from xy data given set of breakpoints.

Usage

hist_xy(xy, breaks)

Arguments

xy Numeric. A data.frame, or object coercible to a data.frame with as.data.frame(), where the columns of the data.frame are, or can be, interpreted as XY coordinates.

breaks List. A list of vectors, one for each dimension of xy, where each vector gives the breakpoints between the histogram cells.

Value

A list with three components:

• x The X dimension breakpoints.
• y The Y dimension breakpoints.
• z A frequency table of the number of xy points in each cell.

Examples

hist_xy(matrix(runif(50), 25), list(seq(0, 1, .2), seq(0, 1, .2)))
variables

FinBIF record variables that can be selected in a finbif occurrence search.

Identifiers

All identifiers are returned in the form of a URI. Identifiers include:

- **record_id** Character. The ID of a record of organism’s occurrence at a time and place.
- **individual_id** Character. ID of an individual organism (e.g., a ringed bird that has been captured multiple times will have a single individual_id and multiple record_ids corresponding to each capture).
- **event_id** Character. Event ID. An event can contain one or more records (e.g., a survey of plants at a particular location and time.)
- **document_id** Character. Document ID. A set of events that share common metadata.
- **form_id** Character. Form ID. The form used to create the document, event, record data.
- **collection_id** Character. Collection ID. All documents, events, and records belong to a collection (e.g., a museum collection, or the datasets collected by a specific institution). Collections themselves can be part of a larger (super)collection (e.g., all the collections at a specific museum). Only the lowest level collection ID for a record is returned. Use finbif_collections() to explore the hierarchy of collections.
- **source_id** Character. Source ID. The source of the collection’s data.

Taxa

Variables related to taxonomy of records include:

- **taxon_id** Character. The taxon ID in the form of a URI.
- **orig_taxon_id** Character. The taxon ID before (if any) annotation.
- **annotated_taxon_id** Character. The new taxon ID if the record has had it’s taxonomy annotated.
- **reported_taxon_id** Character. The taxon ID as originally reported by the record creator.
- **scientific_name** Character. Scientific name of taxon.
- **orig_scientific_name** Character. The scientific name before (if any) annotation.
- **scientific_display_name** Character. Scientific name of taxon formatted for display (e.g., taxa with genus only will be formatted as Genus sp.).
- **orig_scientific_display_name** Character. Scientific display name before (if any) annotation.
- **common_name** Character. Common (vernacular) name of taxon.
- **orig_common_name** Character. Common name before (if any) annotation.
• reported_name Character. The name of the taxon as originally reported by the record creator.
• scientific_name_italicised Logical. Is the scientific name normally italicised (i.e., is the taxonomic rank genus or below.)
• orig_scientific_name_italicised Logical. Is the original scientific name normally italicised.
• scientific_name_author Character. The authority for the taxon scientific name.
• orig_scientific_name_author Character. The authority for the taxon scientific name before (if any) annotation.
• reported_author Character. The authority of the taxon as originally reported by the record creator.
• taxon_rank Character. The taxonomic rank of the taxon (in the form of a URI).
• orig_taxon_rank Character. The taxonomic rank of the taxon (in the form of a URI) before (if any) annotation.
• informal_groups List. The informal taxonomic groups that the taxon belongs to (e.g., birds) in the form of URIs.
• orig_informal_groups List. The informal taxonomic groups that the taxon belonged to before (if any) annotation.
• reported_informal_groups List. The informal taxonomic groups that the taxon belongs to as reported by the record creator.
• taxon_checklist Character. The checklist (as a URI) that that taxon is found in.
• orig_taxon_checklist Character. The checklist (as a URI) that that taxon was found in before (if any) annotation.
• taxon_finnish Logical. Is the taxon considered Finnish. The definition of a Finnish taxon differs by taxonomic group?
• orig_taxon_finnish Logical. Was the taxon considered Finnish before (if any) annotation?
• is_unidentifiable Logical. Is the record unable to be identified?

**Abundance, sex & life history**

Variables related to abundance, sex and life history include:

• abundance Integer. Number of individuals recorded or inferred from the record. Note that many records with abundance = 1 only indicate the record is of one individual and may not necessarily imply that this was the abundance at that specified place and time (e.g., a preserved museum specimen consisting of a single individual).
• {female, male}_abundance Integer. Number of female or male individuals recorded.
• pair_abundance Integer. Number of mating pairs recorded.
• abundance_verbatim Character. The abundance as reported by the record creator.
• life_stage Character. Life stage of individual(s) recorded.
• sex Character. Sex of individual(s) recorded.
variables

Location

Variables related to the location of records include:

- \{lat|lon\}_wgs84 Numeric. Coordinates (in WGS84 coordinate system) of the central point of a bounding box encompassing the record’s geographic coverage.
- \{lat|lon\}_{[min|max]}_{[euref|kkj|wgs84]} Numeric. Vertices of a bounding box encompassing the record’s geographic coverage. Coordinates are available in EUREF, KKJ, or WGS84.
- coordinates_uncertainty Integer. The horizontal distance (in meters) from the record’s given coordinates describing the smallest circle containing the whole of the record’s location.
- coordinates_source
- footprint_{euref|kkj|wgs84} Character. Well-Known Text (WKT) representation of the geographic shape defining the location of the record in either EUREF, KKJ or WGS84 coordinate systems.
- country Character. The country of the record’s location.
- province Character. The administrative area directly below the level of country. For data from Finland FinBIF uses the concept of Biogeographical Province. See link for details.
- municipality Character. Administrative level below province.
- higher_geography Character. Geographic place name that is at higher level than country.
- line_length_m Integer. The length of linear locations (e.g., line transect surveys).
- area_m2 Integer. The size of record’s location in meters squared.
- is_breeding_location Logical. Whether or not the occurrence is recorded at a known breeding location.

Time

Variables related to time of record include:

- date_start Character. The date the recording event began.
- date_end Character. The date the recording event ended.
- hour_start Integer. The hour (24 hour time) of the day the recording event began.
- hour_end Integer. The hour (24 hour time) of the day the recording event ended.
- minute_start Integer. The minute of the hour the recording event began.
- minute_end Integer. The minute of the hour the recording event started.
- ordinal_day_start Integer. The ordinal day of the year the recording event began.
- ordinal_day_end Integer. The ordinal day of the year the recording event ended.
- season_start Integer. The day of the year the recording event began. A four digit number indicating the day of the year in MMDD (%m%d) format.
- season_end Integer. The day of the year the recording event ended. A four digit number indicating the day of the year in MMDD (%m%d) format.
- century Integer. The century during which the recording event occurred (NA if the event spans multiple centuries).
- decade Integer. The decade during which the recording event occurred (NA if the event spans multiple decades).
• year Integer. The year during which the recording event occurred (NA if the event spans multiple years).
• month Integer. The month of the year during which the recording event occurred (NA if the event spans multiple months).
• day Integer. The day of the month during which the recording event occurred (NA if the event spans multiple days).
• formatted_date_time Character. Date and time of the recording event formatted for display.
• date_created Character. The date the original data was created.
• first_load_date Character. The date the record was first loaded into the FinBIF database.
• modified_date Character. The most recent date the original data was modified.
• load_date Character. The most recent date the record was loaded into the FinBIF database.

Data restrictions

Variables related to restricted records include:

• restriction Logical. Has the record been restricted in some way (e.g., geospatially aggregated).
• restriction_level Character. What level of restriction has been applied to the record.
• restriction_reasons List. List of reasons restriction has been applied.
• restriction_reasons_conservation Logical. Has the record been restricted for reasons of conservation.
• restriction_reasons_embargo Logical. Has the record been restricted due to a temporary data embargo.
• restriction_reasons_custom Logical. Has the record been restricted for other reasons.

Data quality

Variables related to the quality of records include:

• any_issues Logical. Are there any data quality issues associated with the record’s event or document.
• record_reliable Logical. The "collection_reliability" is 4-5 or the record has been "confirmed" and has no quality issues (record, event or document).
• collection_reliability Integer. A numeric rating of reliability from 1 (low reliability) to 5 (high reliability) for the collection the record comes from.
• taxon_reliability Character. Reliability of the record taxonomic identification.
• taxon_reliability_message Remarks associated with taxon_reliability.
• taxon_reliability_source Source of the taxon_reliability rating.
• reported_taxon_confidence Reliability of the record taxonomic identification as reported by the original data author.
• {document|time|location|event|record}_issue Character. Issues with record associated with its document, time, location, event, or the record itself.
• {document|time|location|event|record}_issue_message Character. Details about the issue.
• {document|time|location|event|record}_issue_source Source determining the issue.
Misc

Other variables:

- `observers_ids` List. List of observer identifiers for the record.
- `determiner` Character. Person who determined the taxonomic identification of the record.
- `record_basis` Character. The type of or method used to obtain the record.
- `superrecord_basis` Character. Higher level type of or method used to obtain the record.
- `type_specimen` Logical. Whether or not the record is of a type specimen.
- `is_wild` Logical. Whether or not the record is of a "wild" organism.
- `license` Character. The license of the data associated with the record.
- `{document|event|record}_notes` Character. Notes associated with the document, event or record itself.
- `{document|record}_keywords` List. List of keywords associated with the document or record.
- `record_annotation_count` Integer. How many annotations are associated with the record.
- `sample_count` Integer. How many material samples (DNA extractions, etc., ...) are associated with the record.
- `{document|event|record}_media_count` Integer. How many media items (images, audio, video, etc., ...) are associated with the record’s document, event or the record itself.
Index

*Topic datasets
  finland_map, 17

as.data.frame.finbif_records, 2
as.data.frame.finbif_records_list
  (as.data.frame.finbif_records), 2
as_date, 6, 7
breaks_xy, 3

 caching, 4

 Date, 6, 7

 filters, 4
 finbif_check_taxa, 9
 finbif_check_taxa(), 4
 finbif_clear_cache, 10
 finbif_collections, 10
 finbif_informal_groups, 11
 finbif_informal_groups(), 5
 finbif_metadata, 12
 finbif_metadata(), 5
 finbif_occurrence, 13
 finbif_records, 14
 finbif_request_token, 16
 finbif_taxa, 16
 finbif_taxa(), 4
 finland_map, 17

 hist_xy, 18

 Interval, 6

 variables, 19