

Package ‘c14bazAAR’

October 28, 2018

Title Download and Prepare C14 Dates from Different Source Databases

Version 1.0.2

URL <https://github.com/ISAAKiel/c14bazAAR>

Date 2018-09-19

Description

Query different C14 date databases and apply basic data cleaning, merging and calibration steps.

Depends R (>= 3.4.0)

License GPL-2 | file LICENSE

Encoding UTF-8

LazyData true

Imports crayon (>= 1.3.4), data.table (>= 1.11.4), dplyr (>= 0.7.2),
magrittr (>= 1.5), pbapply (>= 1.3-3), RCurl (>= 1.95-4.8),
rlang (>= 0.1.1), tibble (>= 1.3.3), tidyr (>= 0.6.3)

Suggests Bchron (>= 4.2.6), countrycode (>= 1.00.0), dataverse (>=
0.2.0), lwgeom (>= 0.1-4), openxlsx (>= 4.0.17), plyr (>=
1.8.4), rgeos (>= 0.3-26), rworldmap (>= 1.3-6), rworldxtra (>=
1.01), sf (>= 0.5-4), stringdist (>= 0.9.4.6), testthat

RoxygenNote 6.1.0

NeedsCompilation no

Author Clemens Schmid [aut, cre, cph],
Dirk Seidensticker [aut],
Daniel Knitter [aut],
Martin Hinz [aut],
David Matzig [aut],
Wolfgang Hamer [aut],
Kay Schmütz [aut]

Maintainer Clemens Schmid <clemens@nevrome.de>

Repository CRAN

Date/Publication 2018-10-28 22:00:03 UTC

R topics documented:

as.sf	2
c14_date_list	3
calibrate	4
circumference_calculator	5
classify_material	6
coordinate_precision	7
country_thesaurus	7
determine_country_by_coordinate	8
digits_counter	10
duplicates	10
enforce_types	11
example_c14_date_list	12
fuse	12
get_country_thesaurus	13
get_dates	13
get_db_url	14
get_material_thesaurus	14
individual_precision	15
material_thesaurus	15
order_variables	16
variable_reference	16
Index	18

as.sf	<i>Convert a c14_date_list to a sf object</i>
-------	--

Description

Most 14C dates have point position information in the coordinates columns **lat** and **lon**. This allows them to be converted to a spatial simple feature collection as provided by the `sf` package. This simplifies for example mapping of the dates.

Usage

```
as.sf(x)

## Default S3 method:
as.sf(x)

## S3 method for class 'c14_date_list'
as.sf(x)
```

Arguments

x an object of class `c14_date_list`

Value

an object of class sf

Examples

```
sf_c14 <- as.sf(example_c14_date_list)

## Not run:
library(mapview)
mapview(sf_c14$geom)

## End(Not run)
```

c14_date_list	c14_date_list
---------------	----------------------

Description

The **c14_date_list** is the central data structure of the c14bazAAR package. It's a tibble with set of custom methods and variables. Please see `c14bazAAR::variable_reference` for a description of the variables. Further available variables are ignored.

If an object is of class data.frame or tibble (tbl & tbl_df), it can be converted to an object of class **c14_date_list**. The only requirement is that it contains the essential columns **c14age** and **c14std**. The `as` function adds the string "c14_date_list" to the classes vector of the object and applies `order_variables()`, `enforce_types()` and the helper function `clean_latlon()` to it.

Usage

```
as.c14_date_list(x, ...)

is.c14_date_list(x, ...)

## S3 method for class 'c14_date_list'
format(x, ...)

## S3 method for class 'c14_date_list'
print(x, ...)
```

Arguments

x	an object
...	further arguments passed to or from other methods

Examples

```
# c14_date_list can be crafted manually:
as.c14_date_list(data.frame(c14age = c(2000, 2500), c14std = c(30, 35)))

# The c14_date_list class is stripped if
# you apply functions to a c14_date_list
# that return tibbles or data.frames.
# You have to add the class again afterwards:
library(magrittr)
example_c14_date_list %>%
  dplyr::filter(sourcedb == "CALPAL") %>%
  as.c14_date_list()

is.c14_date_list(5) # FALSE
is.c14_date_list(example_c14_date_list) # TRUE

print(example_c14_date_list)
```

calibrate

*Calibrate all valid dates in a **c14_date_list***

Description

Calibrate all dates in a **c14_date_list** with `Bchron::BchronCalibrate()`. The function provides two different kinds of output variables that are added as new list columns to the input **c14_date_list**: **calprobdistr** and **calrange**. **calrange** is accompanied by **sigma**. See `?Bchron::BchronCalibrate` and `?c14bazAAR:::hdr` for some more information.

calprobdistr: The probability distribution of the individual date for all ages with an individual probability $\geq 1e-06$. For each date there's a data.frame with the columns **calage** and **density**.

calrange: The contiguous ranges which cover the probability interval requested for the individual date. For each date there's a data.frame with the columns **dens** and **from** and **to**.

Usage

```
calibrate(x, choices = c("calrange"), sigma = 2, ...)

## Default S3 method:
calibrate(x, choices = c("calrange"), sigma = 2, ...)

## S3 method for class 'c14_date_list'
calibrate(x, choices = c("calrange"),
  sigma = 2, ...)
```

Arguments

x an object of class `c14_date_list`

choices	whether the result should include the full calibrated probability dataframe ('calprobddistr') or the sigma range ('calrange'). Both arguments may be given at the same time.
sigma	the desired sigma value (1,2,3) for the calibrated sigma ranges
...	passed to Bchron::BchronCalibrate()

Value

an object of class `c14_date_list` with the additional columns **calprobddistr** or **calrange** and **sigma**

Examples

```
calibrate(  
  example_c14_date_list,  
  choices = c("calprobddistr", "calrange"),  
  sigma = 1  
)
```

circumference_calculator
circumference_calculator

Description

circumference_calculator

Usage

```
circumference_calculator(x, mode)
```

Arguments

x	vector of latitude or longitude coordinates
mode	a character "lat" or "lon"

Value

vector with circumference values at specific latitudes

classify_material *Apply material classification on a **c14_date_list***

Description

Add column **material_thes** with simplified and unified terms for material categories. The classification is manually curated and therefore maybe not up-to-date. It's stored in `c14bazAAR::material_thesaurus`, but to be more independent of CRAN update cycles the current version of the classification list is downloaded directly from github with `c14bazAAR::get_material_thesaurus()`. With this setup you can also easily apply own thesaurus tables.

Usage

```
classify_material(x,
  material_thesaurus = c14bazAAR::get_material_thesaurus(),
  quiet = FALSE)

## Default S3 method:
classify_material(x,
  material_thesaurus = c14bazAAR::get_material_thesaurus(),
  quiet = FALSE)

## S3 method for class 'c14_date_list'
classify_material(x,
  material_thesaurus = c14bazAAR::get_material_thesaurus(),
  quiet = FALSE)
```

Arguments

`x` an object of class `c14_date_list`
`material_thesaurus` a thesaurus table
`quiet` suppress decision log output

Value

an object of class `c14_date_list` with the additional column **material_thes**

Examples

```
classify_material(
  example_c14_date_list,
  quiet = TRUE
)
```

coordinate_precision *Return coordinate precision according to number of digits in the columns **lat** and **lon** of a **c14_date_list***

Description

The precision of the coordinates for each date vary greatly. `c14bazAAR::coordinate_precision()` calculates the mean of the possible deviation in meters and adds it to the **c14_date_list** with the column **coord_precision**.

Usage

```
coordinate_precision(x)

## Default S3 method:
coordinate_precision(x)

## S3 method for class 'c14_date_list'
coordinate_precision(x)
```

Arguments

x an object of class `c14_date_list`

Value

an object of class `c14_date_list` with the additional column **coord_precision**

country_thesaurus *Country Thesaurus*

Description

A small thesaurus for country names.

Format

A tibble with 2 variables.

- **cor** fixed name
- **var** variations

See Also

Other thesauri: [material_thesaurus](#)

determine_country_by_coordinate

*Functions to improve the country attribution in a **c14_date_list***

Description

c14bazAAR provides several functions to check and improve the spatial attribution of the individual dates in a **c14_date_list** to a country.

c14bazAAR::standardize_country_name() adds column **country_thes** with standardized country names. Most source databases come with a column **country** that contains a character name of the origin country for each date. Unfortunately the different source databases don't rely on a unified naming convention and therefore use various terms to represent the same country (for example: United Kingdom, Great Britain, GB, etc.). This function aims to standardize the country naming scheme. To achieve this, it compares the names to values in an external (countrycode::odelist) and an internal (c14bazAAR::country_thesaurus) reference list. The latter needs manual curation to catch semantic and spelling errors in the source databases.

c14bazAAR::determine_country_by_coordinate() adds the column **country_coord** with standardized country attribution based on the coordinate information of the dates. Due to the inconsistencies in the **country** column in many c14 source databases it's often necessary to rely on the coordinate position (**lat & lon**) for reliable country attribution information.

finalize_country_name() picks the country name in a hierarchical order from the results of c14bazAAR::determine_country_by_coordinate() and c14bazAAR::standardize_country_name() functions, followed by the original input of the database. The result is added to the input date list with the column **country_final**.

finalize_country_name() also calls the other functions c14bazAAR::determine_country_by_coordinate() and c14bazAAR::standardize_country_name() if the necessary columns are missing yet.

Usage

```
determine_country_by_coordinate(x, suppress_spatial_warnings = TRUE)
```

```
## Default S3 method:
determine_country_by_coordinate(x,
  suppress_spatial_warnings = TRUE)
```

```
## S3 method for class 'c14_date_list'
determine_country_by_coordinate(x,
  suppress_spatial_warnings = TRUE)
```

```
finalize_country_name(x, quiet = FALSE)
```

```
## Default S3 method:
finalize_country_name(x, quiet = FALSE)
```

```

## S3 method for class 'c14_date_list'
finalize_country_name(x, quiet = FALSE)

standardize_country_name(x, country_thesaurus = get_country_thesaurus(),
  codesets = c("country.name.de", "iso3c"), quiet = FALSE, ...)

## Default S3 method:
standardize_country_name(x,
  country_thesaurus = get_country_thesaurus(),
  codesets = c("country.name.de", "iso3c"), quiet = FALSE, ...)

## S3 method for class 'c14_date_list'
standardize_country_name(x,
  country_thesaurus = get_country_thesaurus(),
  codesets = c("country.name.de", "iso3c"), quiet = FALSE, ...)

```

Arguments

x	an object of class <code>c14_date_list</code>
suppress_spatial_warnings	suppress some spatial data messages and warnings
quiet	suppress suppress decision log output
country_thesaurus	data.frame with correct and variants of country names
codesets	which country codesets should be searched for in <code>countrycode::codelist</code> beyond country.name.en ? See <code>?countrycode::codelist</code> for more information
...	additional arguments are passed to <code>stringdist::stringdist()</code> . <code>stringdist()</code> is used for fuzzy string matching of the country names in <code>countrycode::codelist</code>

Value

an object of class `c14_date_list` with the additional columns **country_thes**, **country_coord** and/or **country_final**

Examples

```

library(magrittr)
example_c14_date_list %>%
  determine_country_by_coordinate() %>%
  standardize_country_name() %>%
  finalize_country_name()

```

digits_counter	<i>digits_counter</i>
----------------	-----------------------

Description

counts the digits of the given coordinates

Usage

```
digits_counter(x)
```

Arguments

x vector of coordinate values

Value

vector with number of digits

duplicates	<i>Mark and remove duplicates in a c14_date_list</i>
------------	--

Description

Duplicates are found in `c14bazAAR::mark_duplicates()` by comparison of **labnrs**. Only dates with exactly equal **labnrs** are considered duplicates. Duplicate groups are numbered (from 0) and these numbers linked to the individual dates in the new column **duplicate_group**. Duplicates can be removed with `c14bazAAR::remove_duplicates()`.

While `c14bazAAR::mark_duplicates()` finds duplicates, `c14bazAAR::remove_duplicates()` removes them by merging all dates in a **duplicate_group**. All non-equal variables in the duplicate group are turned to NA. A new column **duplicate_remove_log** documents the variety of entries initially provided (and partially lost by this hard merging operation). `c14bazAAR::remove_duplicates()` needs the column **duplicate_group** and calls `c14bazAAR::mark_duplicates()` if it's missing.

Usage

```
mark_duplicates(x)
```

```
## Default S3 method:
```

```
mark_duplicates(x)
```

```
## S3 method for class 'c14_date_list'
```

```
mark_duplicates(x)
```

```
remove_duplicates(x)
```

```
## Default S3 method:
remove_duplicates(x)

## S3 method for class 'c14_date_list'
remove_duplicates(x)
```

Arguments

x an object of class `c14_date_list`

Value

an object of class `c14_date_list` with the additional columns **duplicate_group** or **duplicate_remove_log**

Examples

```
mark_duplicates(example_c14_date_list)

library(magrittr)
example_c14_date_list %>%
  mark_duplicates() %>%
  remove_duplicates()
```

enforce_types	<i>Enforce variable types in a c14_date_list</i>
---------------	---

Description

Enforce variable types in a **c14_date_list** and remove everything that doesn't fit (e.g. text in a number field). See `c14bazAAR::variable_reference()` for a documentation of the variable types. `enforce_types()` is called in `c14bazAAR::as.c14_date_list()`.

Usage

```
enforce_types(x, suppress_na_introduced_warnings = TRUE)

## Default S3 method:
enforce_types(x,
  suppress_na_introduced_warnings = TRUE)

## S3 method for class 'c14_date_list'
enforce_types(x,
  suppress_na_introduced_warnings = TRUE)
```

Arguments

`x` an object of class `c14_date_list`
`suppress_na_introduced_warnings`
 suppress warnings caused by data removal in type transformation due to wrong database entries (such as text in a number column)

Value

an object of class `c14_date_list`

`example_c14_date_list` *Example c14_date_list*

Description

`c14_date_list` with 200 random dates for tests and example code.

Format

A `c14_date_list`. See `variable_reference` for an explanation of the variable meaning.

`fuse` *Fuse multiple **c14_date_lists***

Description

This function combines **`c14_date_lists`** with `dplyr::bind_rows()`. This is not a joining operation and it therefore might introduce duplicates. See `c14bazAAR::mark_duplicates()` and `c14bazAAR::remove_duplicates()` for a way to find and remove them.

Usage

```

fuse(...)

## Default S3 method:
fuse(...)

## S3 method for class 'c14_date_list'
fuse(...)
```

Arguments

`...` objects of class `c14_date_list`

Value

an object of class `c14_date_list`

```
get_country_thesaurus  get_country_thesaurus
```

Description

Download thesaurus and provide it as tibble.

Usage

```
get_country_thesaurus()
```

```
get_dates  Download radiocarbon source databases and convert them to a
           c14_date_list
```

Description

This functions download source databases and adjust their variables to conform to the definition in `c14bazAAR::variable_reference`. That includes renaming and arranging the variables (with `c14bazAAR::order_variables()`) as well as type conversion (with `c14bazAAR::enforce_types()`) – so all the steps undertaken by `as.c14_date_list()`.

All databases require different downloading and data wrangling steps. Therefore there's a custom getter function for each of them.

`get_all_dates()` is a wrapper to download all dates from all databases and `c14bazAAR::fuse()` the results.

Usage

```
get_14SEA(db_url = get_db_url("14SEA"))
get_aDRAC(db_url = get_db_url("aDRAC"))
get_all_dates()
get_AustArch(db_url = get_db_url("AustArch"))
get_CalPal(db_url = get_db_url("CalPal"))
get_CONTEXT(db_url = get_db_url("CONTEXT"))
get_KITEeastAfrica(db_url = get_db_url("KITEeastAfrica"))
get_EUROEVOL(db_url = get_db_url("EUROEVOL"))
get_RADON(db_url = get_db_url("RADON"))
get_RADONB(db_url = get_db_url("RADON-B"))
```

Arguments

`db_url` weblink to c14 archive file. `c14bazaAR::get_db_url()` fetches the current URL of the data source

Examples

```
## Not run:
aDRAC <- get_aDRAC()
all_dates <- get_all_dates()

## End(Not run)
```

<code>get_db_url</code>	<i>get db url</i>
-------------------------	-------------------

Description

Downloads URLs of c14 source databases from a reference table on github.

Usage

```
get_db_url(db_name)
```

Arguments

`db_name` name of the database

<code>get_material_thesaurus</code>	<i>get_material_thesaurus</i>
-------------------------------------	-------------------------------

Description

Download thesaurus and provide it as tibble.

Usage

```
get_material_thesaurus()
```

`individual_precision` *individual_precision*

Description

`individual_precision`

Usage

```
individual_precision(x, mode)
```

Arguments

`x` vector of coordinates (latitude or longitude)
`mode` argument indicating the 'mode' of the coordinates (whether these are lat or lon)

Value

vector with precision in meters

`material_thesaurus` *Material Thesaurus*

Description

A small thesaurus for material classes.

Format

A tibble with 2 variables.

- **cor** fixed name
- **var** variations

See Also

Other thesauri: [country_thesaurus](#)

order_variables	<i>Order the variables in a c14_date_list</i>
-----------------	--

Description

Arrange variables according to a defined order. This makes sure that a **c14_date_list** always appears with the same outline.

A **c14_date_list** has at least the columns **c14age** and **c14std**. Beyond that there's a selection of additional variables depending on the input from the source databases, as a result of the c14bazAAR functions or added by other data analysis steps. This function arranges the expected variables in a distinct, predefined order. Undefined variables are added at the end.

Usage

```
order_variables(x)

## Default S3 method:
order_variables(x)

## S3 method for class 'c14_date_list'
order_variables(x)
```

Arguments

x an object of class c14_date_list

Value

an object of class c14_date_list

variable_reference	<i>Variable Reference</i>
--------------------	---------------------------

Description

The parameter reference list of c14bazAAR: Which variables in a c14_date_list equal the ones in the source databases and what do they mean. Also contains a full list of the variables in the source databases.

Format

A tibble.

- **c14bazAAR** name of variable in c14bazAAR
- **type** data type of variable in R

- **definition** meaning of variable
- **source** is the variable imported (databases) or generated (c14bazAAR)
- ... variables in source databases

Index

as.c14_date_list (c14_date_list), 3
as.sf, 2

c14_date_list, 3
calibrate, 4
circumference_calculator, 5
classify_material, 6
coordinate_precision, 7
country_attribution
 (determine_country_by_coordinate),
 8
country_thesaurus, 7, 15

determine_country_by_coordinate, 8
digits_counter, 10
duplicates, 10

enforce_types, 11
example_c14_date_list, 12

finalize_country_name
 (determine_country_by_coordinate),
 8

format.c14_date_list (c14_date_list), 3
fuse, 12

get_14SEA (get_dates), 13
get_aDRAC (get_dates), 13
get_all_dates (get_dates), 13
get_AustArch (get_dates), 13
get_CalPal (get_dates), 13
get_CONTEXT (get_dates), 13
get_country_thesaurus, 13
get_dates, 13
get_db_url, 14
get_EUROEVOL (get_dates), 13
get_KITEeastAfrica (get_dates), 13
get_material_thesaurus, 14
get_RADON (get_dates), 13
get_RADONB (get_dates), 13

individual_precision, 15
is.c14_date_list (c14_date_list), 3

mark_duplicates (duplicates), 10
material_thesaurus, 7, 15

order_variables, 16

print.c14_date_list (c14_date_list), 3

remove_duplicates (duplicates), 10

standardize_country_name
 (determine_country_by_coordinate),
 8

variable_reference, 16