Package 'ISRaD'

June 1, 2022

Title Tools and Data for the International Soil Radiocarbon Database

Version 2.2.3

Description This is the central location for data and tools for the development, maintenance, analysis, and deployment of the International Soil Radiocarbon Database (ISRaD). ISRaD was developed as a collaboration between the U.S. Geological Survey Powell Center and the Max Planck Institute for Biogeochemistry. This R package provides tools for accessing and manipulating ISRaD data, compiling local data using the ISRaD data structure, and simple query and reporting functions for ISRaD. For more detailed information visit the ISRaD website at: <https://soilradiocarbon.org/>.

Depends R (>= 3.5.0)

Imports readxl,
writexl,
raster,
dplyr (>= 0.8),
tidyr (>= 1.0),
RCurl,
ggplot2,
maps,
httr,
rio,
rworldmap,
sp

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Suggests devtools,
knitr,
rmarkdown,
rgdal

R topics documented:

checkTemplateFiles 2
compile 3
convert_fm_d14c 4
future14C 4

Gaudinski_2001 5
graven 5
ISRaD.extra 5
ISRaD.extra.calc_atm14c 6
ISRaD.extra.Cstocks 7
ISRaD.extra.delta_delta 8
ISRaD.extra.fill_CN 9
ISRaD.extra.fill_coords 9
ISRaD.extra.fill_country 10
ISRaD.extra.fill_dates 11
ISRaD.extra.fill_rc 11
ISRaD.extra.geospatial 12
ISRaD.extra.geospatial.keys 13
ISRaD.extra.norm14c_year 14
ISRaD.flatten 17
ISRaD.getdata 17
ISRaD.read.entry 18
ISRaD.rep.count.all 19
ISRaD.rep.count.frc 20
ISRaD.rep.entry.stats 20
ISRaD.rep.site.map 21
ISRaD.report 21
ISRaD.save.entry 22
QAQC 23

Index **25**

`checkTemplateFiles` *Check ISRaD Template/Info files*

Description

Check that the template information file and the template file match appropriately.

Usage

```
checkTemplateFiles(outfile = "", verbose = TRUE)
```

Arguments

`outfile`	file to dump the output report. Defaults to an empty string that will print to standard output
`verbose`	if TRUE (default) will print output to specified outfile

Details

Used in compile() function, but primarily a development tool

Value

Nothing (run for side effects).

Examples

```
checkTemplateFiles()
```

`compile` *Compile ISRaD data product*

Description

Compiles template files into ISRaD database format.

Usage

```
compile(
  dataset_directory,
  write_report = FALSE,
  write_out = FALSE,
  return_type = c("none", "list")[2],
  checkdoi = FALSE,
  verbose = TRUE
)
```

Arguments

`dataset_directory`	Directory where completed QAQCed template files are stored.
`write_report`	Boolean flag to write a log file of the compilation. File will be in the specified dataset_directory at "database/ISRaD_log.txt". If a file with this name already exists in this directory it will be overwritten.
`write_out`	Set to TRUE to write the compiled database file in .xlsx format in dataset_directory
`return_type`	A string that defines return object. Acceptable values are "none" or "list"; default is "list".
`checkdoi`	Set to FALSE if you do not want to validate DOIs during QAQC. (Warning: time consuming).
`verbose`	Set to TRUE to print results of function to console.

Examples

```
# Load example dataset Gaudinski_2001
entry <- ISRaD::Gaudinski_2001
# Save as .xlsx file
ISRaD.save.entry(
  entry = entry,
  template_file = system.file("extdata", "ISRaD_Master_Template.xlsx", package = "ISRaD"),
  outfile = file.path(tempdir(), "Gaudinski_2001.xlsx")
)
# Compile .xlsx file/s in dataset_directory into ISRaD database object
ISRaD.compiled <- compile(tempdir(),
  write_report = TRUE, write_out = TRUE,
  return_type = "list", checkdoi = FALSE, verbose = TRUE
)
```

convert_fm_d14c *convert_fm_d14c*

Description

convert fraction modern to d14c and d14c to fraction modern

Usage

```
convert_fm_d14c(fm = NA, d14c = NA, obs_date_y, verbose = TRUE)
```

Arguments

fm	fraction modern
d14c	delta 14c in per mille
obs_date_y	year of observation/sample collection
verbose	prints message stating which conversion was performed

Details

Convenience function for radiocarbon unit conversions. Recommended to set verbose = FALSE for batch conversions.

Author(s)

J. Beem-Miller

Examples

```
convert_fm_d14c(fm = 0.97057, obs_date_y = 2005)
convert_fm_d14c(d14c = -35.86611, obs_date_y = 2005)
```

future14C *Future atmospheric 14C dataset for delta-delta calculation*

Description

Data from: Sierra, C. "Forecasting atmospheric radiocarbon decline to pre-bomb values", Radiocarbon, Vol 60, Nr 4, 2018, p 1055.1066 DOI:10.1017/RDC.2018.33

Usage

```
future14C
```

Format

dataframe

Gaudinski_2001 *Gaudinski Harvard Forest example dataset*

Description

Data from Gaudinski, J., 2001, Belowground carbon cycling in three temperate forests of the eastern United States, University of California Irvine, Ph.D. thesis

Usage

```
Gaudinski_2001
```

Format

list

graven *Graven dataset for delta-delta calculation*

Description

Data from Graven et al 2017 https://www.geosci-model-dev.net/10/4405/2017/gmd-10-4405-2017.pdf

Usage

```
graven
```

Format

dataframe

ISRaD.extra *ISRaD.extra*

Description

Fills transformed and geospatial data where possible to generate an enhanced version of ISRaD.

Usage

```
ISRaD.extra(database, geodata_directory)
```

Arguments

database	ISRaD dataset object
geodata_directory	directory where geospatial data are found

Details

Fills fraction modern, delta 14C, delta-delta, profile coordinates, bulk density, organic C concentration, and SOC stocks from entered data; fills soil and climatic data from external geospatial data products

Value

New ISRaD_extra object with derived, transformed, and filled columns.

Examples

```
# Load example dataset Gaudinski_2001
database <- ISRaD::Gaudinski_2001
# Fill ISRaD.extra data
database.x <- ISRaD.extra(database,
  geodata_directory = system.file("extdata", "geodata_directory", package = "ISRaD")
)
```

`ISRaD.extra.calc_atm14c`

ISRaD.extra.calc_atm14c

Description

Calculates atmospheric 14c in the year of sampling for each record in an ISRaD object

Usage

```
ISRaD.extra.calc_atm14c(database, future = TRUE)
```

Arguments

`database`	ISRaD object
`future`	Project atmospheric radiocarbon into the future?

Details

Creates new column for atmospheric 14c (xxx_atm14c). Observation year and profile coordinates must be filled (use ISRaD.extra.fill_dates, and ISRaD.extra.fill_coords functions). The relevant atmospheric 14c data (northern or southern hemisphere or tropics) are determined by profile coordinates. Projection for 2016 to 2021 uses the four quarter average projected atmospheric radiocarbon concentration for Central Europe as estimated in Sierra (2019).

Notes: Central Europe projection (Sierra, 2019) used for northern hemisphere samples as these projections perform better against observations than northern hemisphere projection; southern hemisphere and tropic atmospheric radiocarbon projection are lagged by 2.5 per mille, as this is the mean lag observed from 2000 to 2015 in the Graven (2017) dataset.

Value

ISRaD_data object with new atmospheric zone and atmospheric 14c columns in relevant tables.

Author(s)

J. Beem-Miller and C. Hicks-Pries

References

Graven et al. 2017. Compiled records of carbon isotopes in atmospheric CO2 for historical simulations in CMIP6. Geosci. Model Dev., 10: 4405–4417 https://www.geosci-model-dev.net/10/4405/2017/gmd-10-4405-2017.pdf

Sierra, C. 2018. Forecasting atmospheric radiocarbon decline to pre-bomb values. Radiocarbon, 60(4): 1055-1066 https://doi.org/10.1017/RDC.2018.33

Examples

```
# Load example dataset Gaudinski_2001
database <- ISRaD::Gaudinski_2001
# Fill profile coordinates
database.x <- ISRaD.extra.fill_coords(database)
# Fill dates
database.x <- ISRaD.extra.fill_dates(database.x)
# Fill delta 14C from fraction modern
database.x <- ISRaD.extra.fill_rc(database.x)
# Fill atmospheric 14c
database.x <- ISRaD.extra.calc_atm14c(database.x)
```

`ISRaD.extra.Cstocks` *ISRaD.extra.Cstocks*

Description

Calculates soil organic carbon stock

Usage

```
ISRaD.extra.Cstocks(database)
```

Arguments

`database`	ISRaD dataset object.

Details

Function first fills lyr_bd_samp, lyr_c_org, lyr_c_org, lyr_coarse_tot. Notes:

1) SOC stocks can only be calculated if organic carbon concentration and bulk density data are available

2) SOC stocks are calculated for the fine earth fraction (<2mm).

Value

ISRaD_data object with filled columns "lyr_coarse_tot_filled", "lyr_bd_samp_filled", "lyr_c_inorg_filled", "lyr_c_org_filled", "lyr_soc_filled".

Author(s)

J. Beem-Miller

Examples

```
# Load example dataset Gaudinski_2001
database <- ISRaD::Gaudinski_2001
database.x <- ISRaD.extra.Cstocks(database)
```

`ISRaD.extra.delta_delta` *ISRaD.extra.delta_delta*

Description

Calculates the difference between sample delta 14c and the atmosphere for the year of collection (delta-delta)

Usage

```
ISRaD.extra.delta_delta(database)
```

Arguments

`database`	ISRaD dataset object

Details

Creates new column for delta-delta value. Function "ISRaD.extra.calc_atm14c" should be run first.

Value

ISRaD_data object with new delta-delta columns in relevant tables.

Author(s)

J. Beem-Miller

Examples

```
# Load example dataset Gaudinski_2001
database <- ISRaD::Gaudinski_2001
# Fill profile coordinates
database.x <- ISRaD.extra.fill_coords(database)
# Fill dates
database.x <- ISRaD.extra.fill_dates(database.x)
# Fill delta 14C from fraction modern
database.x <- ISRaD.extra.fill_rc(database.x)
# Fill atmospheric 14c
database.x <- ISRaD.extra.calc_atm14c(database.x)
# Fill delta delta
database.x <- ISRaD.extra.delta_delta(database.x)
```

ISRaD.extra.fill_CN *ISRaD.extra.fill_c_to_n*

Description

Calculates missing C:N ratios for records with reported C and N data

Usage

```
ISRaD.extra.fill_CN(database)
```

Arguments

database	ISRaD dataset object

Details

When possible, missing C:N ratios are calculated for records in the layer and fraction tables using reported values for organic C and total N. Variable "lyr_c_org_filled" must exist for function to work on layer table data. If you are running the function on a standard ISRaD database object (i.e. NOT ISRaD_extra) it is recommended to run the function "ISRaD.extra.Cstocks" first in order to create and fill the required "lyr_c_org_filled" column.

Value

ISRaD database object with gap-filled C:N data in new column ""

Author(s)

Shane Stoner & J. Beem-Miller

Examples

```
# Load example dataset Gaudinski_2001
database <- ISRaD::Gaudinski_2001
# Create lyr_c_org_filled column
database.x <- ISRaD.extra.Cstocks(database)
# Fill CN data
database.x <- ISRaD.extra.fill_CN(database.x)
```

ISRaD.extra.fill_coords *ISRaD.extra.fill_coords*

Description

Fills profile coordinates from site coordinates if profile coordinates not reported.

Usage

```
ISRaD.extra.fill_coords(database)
```

Arguments

database	ISRaD dataset object.

Value

ISRaD_data object with filled profile coordinates.

Author(s)

J. Beem-Miller

Examples

```
# Load example dataset Gaudinski_2001
database <- ISRaD::Gaudinski_2001
# Fill profile coordinates
database.x <- ISRaD.extra.fill_coords(database)
```

ISRaD.extra.fill_country

ISRaD.extra.fill_country

Description

Fills country code from profile coordinates

Usage

```
ISRaD.extra.fill_country(database, continent = FALSE, region = FALSE)
```

Arguments

database	ISRaD dataset object.
continent	Boolean noting whether a column should be added for extracted continent (6 continent model: "Eurasia")
region	Boolean noting whether a column should be added for extracted region (7 continent model: "Europe", "Asia")

Value

ISRaD_data object with extracted country names.

Author(s)

Shane Stoner & J. Beem-Miller

Examples

```
# Load example dataset Gaudinski_2001
database <- ISRaD::Gaudinski_2001
# Fill profile coordinates
database.x <- ISRaD.extra.fill_coords(database)
# Fill country
database.x <- ISRaD.extra.fill_country(database.x)
```

ISRaD.extra.fill_dates *ISRaD.extra.fill_dates*

Description

Fills frc_obs_date_y and inc_obs_date_y columns from lyr_obs_date_y if not reported.

Usage

```
ISRaD.extra.fill_dates(database)
```

Arguments

database	ISRaD dataset object.

Details

QAQC does not require frc_obs_date_y or inc_obs_date_y fields to be filled in. Therefore it is recommended to run this function prior to running the functions "ISRaD.extra.fill_rc" and "ISRaD.extra.calc_atm14c", which require xxx_obs_date_y data.

Value

ISRaD_data object with filled frc_obs_date_y and inc_obs_date_y fields.

Examples

```
# Load example dataset Gaudinski_2001
database <- ISRaD::Gaudinski_2001
# Fill dates
database.x <- ISRaD.extra.fill_dates(database)
# Fraction table now has lyr_obs_date_y values in frc_obs_date_y field
```

ISRaD.extra.fill_rc *ISRaD.extra.fill_14c*

Description

Fills delta 14c or fraction modern data if either are missing

Usage

```
ISRaD.extra.fill_rc(database)
```

Arguments

database	ISRaD dataset object.

Details

Warning: xxx_obs_date_y columns must be filled for this to work! This function also fills standard deviation and sigma values. Note that this function replaces two older functions ("ISRaD.extra.fill_fm" and "ISRaD.extra.fill_14c") from ISRaD v1.0 that did not work properly.

Value

ISRaD_data object with filled radiocarbon data columns in all tables

Author(s)

J. Beem-Miller & A. Hoyt

References

Stuiver and Polach, 1977

Examples

```
# Load example dataset Gaudinski_2001
database <- ISRaD::Gaudinski_2001
# Note that some flx_14c values are NA
is.na(database$flux$flx_14c)
is.na(database$layer$lyr_14c)
# Fill dates
database.x <- ISRaD.extra.fill_dates(database)
# Fill rc values
database.x <- ISRaD.extra.fill_rc(database.x)
# Missing radiocarbon data has now been filled if possible, e.g. column flx_14c in the "flux" table
is.na(database$flux$flx_14c)
```

ISRaD.extra.geospatial

ISRaD.extra.geospatial

Description

Extracts data from a user-supplied raster file and adds data as a new variable at the profile level

Usage

```
ISRaD.extra.geospatial(
  database,
  geodata_directory,
  CRS = "+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"
)
```

Arguments

database	ISRaD dataset object
geodata_directory	Directory where geospatial data are found
CRS	Coordinate reference system used for geospatial datasets

Details

Generic function that uses geographic coordinates of profiles to extract data from one or more raster files. Raster data will be added as new variables at the profile level.

The new variable name will be a concatenation of "pro_", and the file name (excluding the file extension). The ISRaD file name convention for geospatial files uses a 6 component string, separated by "_". Missing components can be replaced with "x" ("x"s will be dropped before creating variable names). The 6 components are as follows:

1) Short description of the data type, e.g. "bd" for bulk density
2) Top layer depth or exact depth (numeric, cm)
3) Bottom layer depth (numeric, cm)
4) Year of data observation (numeric)
5) Data units (e.g. mmyr for mean annual precipitation)
6) Any relevant notes

Coordinate reference system can be specified with the "CRS" argument; default is WGS84. Note that all files in geodata_directory must use the same CRS.

Value

Updated ISRaD_extra object with new columns at the profile level

Examples

```
# Load example dataset Gaudinski_2001
database <- ISRaD::Gaudinski_2001
# Fill profile coordinates
database <- ISRaD.extra.fill_coords(database)
# Run function
# Note that geospatial data in pkg is only for the Gaudinski_2001 dataset
# Users may supply their own geospatial data as long as it can be read by the raster package
database.x <- ISRaD.extra.geospatial(database,
  geodata_directory = system.file("extdata", "geodata_directory", package = "ISRaD")
)
```

`ISRaD.extra.geospatial.keys`

ISRaD.extra.geospatial.keys

Description

Recode numeric values from categorical geospatial data products

Usage

```
ISRaD.extra.geospatial.keys(database, geodata_keys)
```

Arguments

database	ISRaD dataset object
geodata_keys	directory where geospatial data are found

Details

Generic function that reads .csv files paired with categorical raster data and recodes extracted data in the ISRaD_extra object. For the function to work, the .csv filenames must be identical to the corresponding raster filenames (except for the file extension). Additionally, the first column of the .csv file must contain the numeric identifier and the second column the corresponding character value.

Value

Updated ISRaD_extra object with recoded columns.

Examples

```
# Load example dataset Gaudinski_2001
database <- ISRaD::Gaudinski_2001
# Fill profile coordinates
database <- ISRaD.extra.fill_coords(database)
# Fill geospatial data
database.x <- ISRaD.extra.geospatial(database,
  geodata_directory = system.file("extdata", "geodata_directory", package = "ISRaD")
)
# Recode numeric data to categorical
database.x <- ISRaD.extra.geospatial.keys(database.x,
  geodata_keys = system.file("extdata", "geodata_keys", package = "ISRaD")
)
```

ISRaD.extra.norm14c_year

ISRaD.extra.norm14c_year

Description

Normalizes delta 14c values to a given year (norm_year)

Usage

```
ISRaD.extra.norm14c_year(
  obs_d14c,
  obs_year,
  atm_zone,
  norm_year,
  df,
  slow = TRUE,
  tau = FALSE,
  verbose = FALSE
)
```

Arguments

obs_d14c	column name in df with observed delta 14c values to be normalized OR numeric value
obs_year	column name in df with year in which obs_d14c was observed (sample collection year) OR numeric value
atm_zone	column name in df with atmospheric zone for obs_d14c OR character string. Notes: column values/character string must be one of c("NHc14", "SHc14", "Tropicsc14"). "NHc14" = > 30 degrees latitude; "SHc14" = < -30 latitude; "Tropicsc14" = < 30 & > -30 degrees latitude.
norm_year	desired normalization year (numeric)
df	data frame with columns for observed d14c (obs_d14c), observation year (obs_year), and atmospheric zone (atm_zone)
slow	if TRUE (default) normalized 14c value will be fit using the slower solution for tau
tau	if TRUE, the solution for tau will be returned along with the normalized 14c value (default = FALSE)
verbose	Show progress bar? TRUE/FALSE (default = FALSE)

Details

This function can be run to return either a column of normalized 14c values in the supplied data frame, or a single normalized 14c value. For the data frame method, the inputs 'obs_d14c', 'obs_year', and 'atm_zone' should correspond to column names in the data frame (see Example 1). For the single value method, the inputs for 'obs_d14c', 'obs_year', and 'atm_zone' are single values (Example 2).

The function works by creating a one pool steady-state model using atmospheric 14c over the period 1850 to 2021. Turnover time (tau^-1) is determined by fitting the model to the observed delta 14c (obs_d14c) in the observation year (obs_year), and the normalized 14c value is calculated by running the model forwards or backwards to the desired normalization year (norm_year). Note that highly negative values of delta 14c (e.g. < -100) are unaffected by normalization.

The curvature of the bomb peak can lead to two viable solutions for tau in a one pool model. Determining which value is more appropriate depends on the carbon dynamics of the system, and cannot be determined a priori (Trumbore 2000). The 'slow' parameter can be used to select which tau is used for calculating the normalized 14c value.

In certain cases the algorithm used to determine tau will fail to find a solution. This situation arises when the observed radiocarbon value is too high relative to the year of observation. This problem is well documented (Gaudinski et al. 2000), and can be solved by introducing a time-lag for the carbon inputs to the system. However, this functionality is beyond the scope of this function. In order to prevent the algorithm from being caught in an infinite loop, the function terminates the search for tau when the predicted delta 14c value drops below -100 per mille. Accordingly, the values estimated for normalized 14c in these cases will be approximately -100 per mille. In general, samples where the algorithm failed to converge can be identified in a larger dataset by searching for records with observed 14c values > 0 and normalized 14c values < 0.

Example 1 shows how to run the function when the 'df' argument corresponds to a table from an ISRaD object, e.g. "flux", "layer", etc.
Example 2 shows how to run the function when single values are supplied and 'df' is absent.

Note: There is no guarantee that normalized 14c values will be meaningful as the model assumes a well-mixed homogenous system, and this is rarely the case in soils.

Value

data frame with normalized 14c values in new column OR single normalized 14c value

Author(s)

J. Beem-Miller and J. Randerson

References

Gaudinski et al. 2000. Soil carbon cycling in a temperate forest: radiocarbon-based estimates of residence times, sequestration rates and partitioning of fluxes. Biogeochemistry 51: 33–69 https://doi.org/10.1023/A:1006301010014

Trumbore, S. 2000. Age of Soil Organic Matter and Soil Respiration: Radiocarbon Constraints on Belowground C Dynamics. Ecological Applications, 10(2): 399–411 http://dx.doi.org/10.2307/2641102

Examples

```
# Load example dataset Gaudinski_2001
database <- ISRaD::Gaudinski_2001
# Fill profile coordinates
database.x <- ISRaD.extra.fill_coords(database)
# Fill dates
database.x <- ISRaD.extra.fill_dates(database.x)
# Fill delta 14C from fraction modern
database.x <- ISRaD.extra.fill_rc(database.x)
# Fill atmospheric 14c
database.x <- ISRaD.extra.calc_atm14c(database.x)
# Run normalization function for the year 2010 with layer data
# Example 1
database.x$layer <- ISRaD.extra.norm14c_year(
  obs_d14c = "lyr_14c",
  obs_year = "lyr_obs_date_y",
  atm_zone = "pro_graven_zone",
  norm_year = 2010,
  df = database.x$layer,
  verbose = TRUE
)
# Example 2
ISRaD.extra.norm14c_year(
  obs_d14c = 182.8958,
  obs_year = 1996,
  atm_zone = "NHc14",
  norm_year = 2010
)
```

`ISRaD.flatten` *ISRaD.flatten*

Description

Joins tables in ISRaD based on linking variables and returns flat data frame/s

Usage

```
ISRaD.flatten(database, table)
```

Arguments

`database`	ISRaD dataset object: e.g. ISRaD_data, or ISRaD_extra
`table`	ISRaD table of interest ("flux", "layer", "interstitial", "fraction", "incubation"). Must be entered with "".

Details

ISRaD.extra.flatten generates 2-dimensional matrices for user-specified ISRaD tables by joining higher level tables (metadata, site, profile, layer) to lower level tables (layer, fraction, incubation, flux, interstitial).

Value

A dataframe with nrow = nrow(table) and ncol = sum(ncol(meta), ncol(site), ncol(profile), ..., ncol(table))

Author(s)

J. Beem-Miller

Examples

```
# Load example dataset Gaudinski_2001
database <- ISRaD::Gaudinski_2001
fractions <- ISRaD.flatten(database, "fraction")
layers <- ISRaD.flatten(database, "layer")
```

`ISRaD.getdata` *ISRaD.getdata*

Description

Retrieves most recent version of ISRaD data from github

Usage

```
ISRaD.getdata(
  directory,
  dataset = "full",
  extra = FALSE,
  force_download = FALSE
)
```

Arguments

directory	Location of ISRaD_database_files folder. If not found, it will be created.
dataset	Specify which data you want. Options are c("full", flux","interstitial","incubation","fraction","layer")
extra	TRUE/FALSE. If TRUE, the ISRaD_extra object will be returned. If FALSE, ISRaD_data will be returned. Default is FALSE.
force_download	TRUE/FALSE. If ISRaD_database files already exist in the specified directory, new data will not be downloaded by default. If force_download is set to TRUE, the newest data from github will be downloaded and overwrite any existing files.

Value

ISRaD data object

Examples

```
# Return full dataset ("full")
ISRaD_full <- ISRaD.getdata(tempdir(), dataset = "full", extra = FALSE)
# Return full dataset plus "extra" filled data
ISRaD_extra <- ISRaD.getdata(tempdir(), dataset = "full", extra = TRUE)
# Return only fraction data, including filled fraction data
ISRaD_fractions <- ISRaD.getdata(tempdir(), dataset = "fraction", extra = TRUE)
```

ISRaD.read.entry *ISRaD.read.entry*

Description

Reads ISRaD data object from Excel file in standard template format

Usage

```
ISRaD.read.entry(
  entry,
  template_file = system.file("extdata", "ISRaD_Master_Template.xlsx", package =
    "ISRaD")
)
```

Arguments

entry	ISRaD data object.
template_file	Directory path and name of template file to use (defaults to the ISRaD_Master_Template file built into the package).

Author(s)

J. Beem-Miller

Examples

```
# Load example dataset Gaudinski_2001
entry <- ISRaD::Gaudinski_2001
ISRaD.save.entry(
  entry = entry,
  template_file = system.file("extdata", "ISRaD_Master_Template.xlsx", package = "ISRaD"),
  outfile = file.path(tempdir(), "Gaudinski_2001.xlsx")
)
# Read in .xlsx file
ISRaD.read.entry(file.path(tempdir(), "Gaudinski_2001.xlsx"))
```

`ISRaD.rep.count.all` *ISRaD.rep.count.all*

Description

Generates a report of counts of observations at each level of the database

Usage

```
ISRaD.rep.count.all(database)
```

Arguments

`database`	ISRaD data object

Value

A tibble of observation counts, one column for each database table.

Examples

```
# Load example dataset Gaudinski_2001
database <- ISRaD::Gaudinski_2001
ISRaD.rep.count.all(database)
```

ISRaD.rep.count.frc *ISRaD.rep.count.frc*

Description

Generates a report of fraction level observations, including fraction scheme and properties. Note that this function only counts rows, not 14C observations.

Usage

```
ISRaD.rep.count.frc(database)
```

Arguments

database	ISRaD data object

Examples

```
# Load example dataset Gaudinski_2001
database <- ISRaD::Gaudinski_2001
ISRaD.rep.count.frc(database)
```

ISRaD.rep.entry.stats *ISRaD.rep.entry.stats*

Description

Generates a report of metadata statistics for all entries

Usage

```
ISRaD.rep.entry.stats(database)
```

Arguments

database	ISRaD data object

Examples

```
# Load example dataset Gaudinski_2001
database <- ISRaD::Gaudinski_2001
ISRaD.rep.entry.stats(database)
```

ISRaD.rep.site.map *ISRaD.rep.site.map*

Description

Generates a world map showing locations of all ISRaD sites

Usage

```
ISRaD.rep.site.map(database)
```

Arguments

database	ISRaD data object

Examples

```
# Obtain current ISRaD data
database <- ISRaD.getdata(tempdir(), dataset = "full", extra = FALSE)
# Generate a map of all ISRaD sites
ISRaD.rep.site.map(database)
```

ISRaD.report *ISRaD.report*

Description

Generate basic summary reports of ISRaD data

Usage

```
ISRaD.report(database, report)
```

Arguments

database	ISRaD data object
report	Parameter to define which type of report is desired. The default is "count.all" other options include "entry.stats", "count.frc", or "site.map".

Details

Wrapper for the simple reporting functions ISRaD.rep.count.all, ISRaD.rep.count.frc, ISRaD.rep.entry.stats, ISRaD.rep.site.map

Examples

```
# Load example dataset Gaudinski_2001
database <- ISRaD::Gaudinski_2001
# Report metadata statistics
ISRaD.report(database, report = "entry.stats")
# Report summary statistics for all levels of the database
ISRaD.report(database, report = "count.all")
# Generate a map of all ISRaD sites
ISRaD.report(database, report = "site.map")
```

`ISRaD.save.entry` *ISRaD.save.entry*

Description

Saves ISRaD data object to .xlsx file

Usage

```
ISRaD.save.entry(
  entry,
  template_file = system.file("extdata", "ISRaD_Master_Template.xlsx", package =
    "ISRaD"),
  outfile
)
```

Arguments

`entry`	ISRaD data object
`template_file`	Directory path and name of template file to use (defaults to the ISRaD_Master_Template file built into the package). Not recommended to change this.
`outfile`	File name and path for .xlsx output

Details

This function can be used to save a single entry (or a compiled database in the standard template format) to an .xlsx file.

Note: Replaces the function "ISRaD.save.xlsx" as that function depended on the package openxlsx, which was unstable at the time. This a simpler function and does not maintain the formatting of the template file. The code for the original function is available in the ISRaD github repository in the devScripts directory.

Author(s)

J. Beem-Miller

Examples

```
# Load example dataset Gaudinski_2001
entry <- ISRaD::Gaudinski_2001
ISRaD.save.entry(
  entry = entry,
  template_file = system.file("extdata", "ISRaD_Master_Template.xlsx", package = "ISRaD"),
  outfile = file.path(tempdir(), "Gaudinski_2001.xlsx")
)
```

QAQC *QAQC*

Description

Checks template files for data coherence, formatting, and data entry errors

Usage

```
QAQC(
  file,
  writeQCreport = FALSE,
  outfile_QAQC = "",
  summaryStats = TRUE,
  dataReport = FALSE,
  checkdoi = TRUE,
  verbose = TRUE,
  local = TRUE
)
```

Arguments

file	File path for template file to be checked
writeQCreport	If TRUE, a text report of the QC output will be written to the outfile. Default is FALSE
outfile_QAQC	Filename of the output file (if writeQCreport is TRUE). Default is NULL, with the outfile being written to the directory where the template file is stored and named according to the file being checked.
summaryStats	Prints summary statistics. Default is TRUE.
dataReport	Prints list structure of database. Default is FALSE.
checkdoi	Set to FALSE if you do not want the QAQC check to validate DOIs (if TRUE this will be time consuming). Default is TRUE.
verbose	Set to TRUE to print results of function to console. Default is TRUE.
local	Set to FALSE to fetch most up-to-date template and template info files. If TRUE, the local files or files from CRAN package will be used. Default is TRUE.

Details

This function can also be called from the ISRaD website.

Examples

```
# Load example dataset Gaudinski_2001
entry <- ISRaD::Gaudinski_2001
# Save as .xlsx file
ISRaD.save.entry(
  entry = entry,
  template_file = system.file("extdata", "ISRaD_Master_Template.xlsx", package = "ISRaD"),
  outfile = file.path(tempdir(), "Gaudinski_2001.xlsx")
)
# Run QAQC
QAQC(file.path(tempdir(), "Gaudinski_2001.xlsx"))
```

Index

∗Topic **datasets**
    future14C, 4
    Gaudinski_2001, 5
    graven, 5

checkTemplateFiles, 2
compile, 3
convert_fm_d14c, 4

future14C, 4

Gaudinski_2001, 5
graven, 5

ISRaD.extra, 5
ISRaD.extra.calc_atm14c, 6
ISRaD.extra.Cstocks, 7
ISRaD.extra.delta_delta, 8
ISRaD.extra.fill_CN, 9
ISRaD.extra.fill_coords, 9
ISRaD.extra.fill_country, 10
ISRaD.extra.fill_dates, 11
ISRaD.extra.fill_rc, 11
ISRaD.extra.geospatial, 12
ISRaD.extra.geospatial.keys, 13
ISRaD.extra.norm14c_year, 14
ISRaD.flatten, 17
ISRaD.getdata, 17
ISRaD.read.entry, 18
ISRaD.rep.count.all, 19
ISRaD.rep.count.frc, 20
ISRaD.rep.entry.stats, 20
ISRaD.rep.site.map, 21
ISRaD.report, 21
ISRaD.save.entry, 22

QAQC, 23