

standardizeSnapshot

December 12, 2022

create_logger	<i>Create logger</i>
---------------	----------------------

Description

Initializes a logger with a given logfile.

Usage

```
create_logger(my_logfile)
```

Arguments

my_logfile	path to the log file (character).
------------	-----------------------------------

Value

Returns a logger and creates a logfile at the given path. If the path given does not exist, also creates this path.

get_final_filename	<i>Get final filemane</i>
--------------------	---------------------------

Description

Return the filename from the file columns.

Usage

```
get_final_filename(df)
```

Arguments

df	The dataframe to be copied. Must have columns locationID, season, roll.
----	---

Value

The filename for this file in the format locationID_Sseason_Rroll.csv It there are several locationID, seasons or rolls, they are separated by a dash in the filename: locationID1-locationID2...

get_logfile_name	<i>Name logfile</i>
------------------	---------------------

Description

Create a name for the logfile from an input file/folder.

Usage

```
get_logfile_name()
```

Value

A string with format "log__YYYY-MM-DD_HH:MM:SS.log" where YYYY-MM-DD_HH:MM:SS is the current date/time.

guess_classifier	<i>Guess classifier</i>
------------------	-------------------------

Description

Guesses the classifier used to annotate the data based on the column names given in colnames_df.

Usage

```
guess_classifier(colnames_df)
```

Arguments

colnames_df A character vector of column names.

Value

The classifier: either zooniverse, digikam or traptagger.

`PEcAn.standardizeSnapshot`

standardizeSnapshot is a R package to standardize camera trap records files from the Snapshot Safari project.

Description

Snapshot Safari data comes in different standards, following the method that was used to classify pictures (Zooniverse, TrapTagger or Digikam). This package allows to standardize all data sources to a unique file format and then cleans the files to homogenize records. See Snapshot Safari: a large-scale collaborative to monitor Africa's remarkable biodiversity (Pardo et al, 2021) (<https://www.sajs.co.za/article/view/8134>)

`read_snapshot_files` *Read Snapshot files*

Description

Reads files from a vector of folders (and optionnally ignores some file/folders) into a list of dataframes.

Usage

```
read_snapshot_files(input, except)
```

Arguments

<code>input</code>	a character vector of valid paths: can be files or folders, or a mix of both
<code>except</code>	files to ignore (optional)

Value

A named list of dataframe. Each element of the list is a dataframe containing the contents of a file read from the files list given in input. The names of the list are the file names from the root of input: If the input is a file, it is a filename. If input is a folder, it is the relative path from input to the file inside input.

rename_standard	<i>Rename columns according to standard</i>
-----------------	---

Description

This function renames the existing columns in a dataframe to match the standard names.

Usage

```
rename_standard(
  df,
  classifier = c("zooniverse", "traptagger", "digikam"),
  standard_colnames
)
```

Arguments

df	The dataframe with the columns to rename
classifier	The classifier used to create the dataframe df. Can be 'zooniverse', 'traptagger', 'digikam'.
standard_colnames	A dataframe with 2 columns (at least) named like the classifier and 'new'. The column named like the classifier contains column names that are expected in the initial file. These names will be matched in the column names of df using partial matching (case insensitive and removing blanks). The column 'new' contains the column names that are expected in the final file. Columns in the classifier column will be renamed following the name of the corresponding value in 'new'. If no old column corresponds to 'new' (indicated with a NA) then the column will be created and filled with NAs.

Value

Returns a dataframe for which the columns have been renamed

standardize_snapshot_df	<i>Standardize dataframe</i>
-------------------------	------------------------------

Description

Standardizes a dataframe to the Snapshot standard.

Usage

```
standardize_snapshot_df(df, standard_df, locationID_digikam, classifier)
```

Arguments

df	The dataframe to standardize. It is expected to match the data format for either Zooniverse, TrapTagger or Digikam processed data (i.e. have column names defined in standard_df).
standard_df	The standard dataframe to match column names to the new standard. A dataframe with ≥ 2 columns, one of which must be named 'zooniverse', 'digikam' or 'traptagger' and another one must be named 'new'.
locationID_digikam	Optional locationID to be used for Digikam data (will display a warning if not provided for Digikam data.)
classifier	Optional character for the classifier.

Value

The standardized dataframe: it has the same columns as specified in standard_df\$new, dates and times are standardized to "YYYY-MM-DD" and "HH:MM:SS", capture info (locationID, cameraID, roll, capture and season if classifier is zooniverse) is filled. Columns are also in the same order as provided in standard_df\$new and the rows are ordered by camera, date and time.

standardize_snapshot_list

Standardize a list of dataframes

Description

Standardizes a list of dataframes to the Snapshot standard.

Usage

```
standardize_snapshot_list(df_list, standard_df, classifier)
```

Arguments

df_list	a list of dataframes.
standard_df	The standard dataframe to match column names to the new standard. A dataframe with ≥ 2 columns, one of which must be named 'zooniverse', 'digikam' or 'traptagger' and another one must be named 'new'.
classifier	Optional character or vector of characters for the classifier.

Value

The list of standardized dataframes: each dataframe has the same columns as specified in standard_df\$new, dates and times are standardized to "YYYY-MM-DD" and "HH:MM:SS", capture info (locationID, cameraID, roll, capture and season if classifier is zooniverse) is filled. Columns are also in the same order as provided in standard_df\$new and the rows are ordered by camera, date and time.

standardize_species	<i>Standardize species</i>
---------------------	----------------------------

Description

Eliminate species duplicate names (things like 'birdofprey' and 'birdsofprey')

Usage

```
standardize_species(species)
```

Arguments

species	vector of species names
---------	-------------------------

Value

the vector of species names with names standardized

write_log_message	<i>Write log message</i>
-------------------	--------------------------

Description

Writes a log message. If a logger is provided, writes to that logger; if it is NA, displays a message.

Usage

```
write_log_message(message, logger = NA, level = "info")
```

Arguments

message	The message to display/write to the logger
logger	Logger to write to (log4r object of class "logger") (defaults to NA)
level	Logging level: either 'info', 'warn', 'debug' or 'error'.

Value

Either a message or writes a log (with the logger parameters)

`write_standardized_file`*Write the standardized file*

Description

Write the standardized file

Usage

```
write_standardized_file(df, in_filename, to)
```

Arguments

<code>df</code>	The standardized file
<code>in_filename</code>	The path to the original file. If it has subfolders, the subfolder structure is copied in to.
<code>to</code>	The target folder to copy data in. It must exist.

Value

Writes the file to the folder `to/xxx` where `xxx` is the subdirectory in which the original file was in. Also returns the path to the file.

Index

`create_logger`, [1](#)

`get_final_filename`, [1](#)
`get_logfile_name`, [2](#)
`guess_classifier`, [2](#)

`PEcAn.standardizeSnapshot`, [3](#)

`read_snapshot_files`, [3](#)
`rename_standard`, [4](#)

`standardize_snapshot_df`, [4](#)
`standardize_snapshot_list`, [5](#)
`standardize_species`, [6](#)

`write_log_message`, [6](#)
`write_standardized_file`, [7](#)