Package: forcis (via r-universe)

September 14, 2024

Type Package

Title An R Client to Access the FORCIS Database

Version 0.0.0.9000

Description Provides an interface to the FORCIS database (<https://zenodo.org/doi/10.5281/zenodo.7390791>) on global foraminifera distribution. This package allows to download and to handle FORCIS data. It is part of the FRB-CESAB working group FORCIS. <https://www.fondationbiodiversite.fr/en/the-frb-in-action/programs-and-projects/le-cesab/forcis/>.

URL https://frbcesab.github.io/forcis

BugReports https://github.com/FRBCesab/forcis/issues

License GPL (>= 2)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

VignetteBuilder knitr

Depends R (>= 2.10)

Imports dplyr, ggplot2, jsonlite, rlang, sf, tidyr, utils, vroom

Suggests fs, knitr, rmarkdown, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Repository https://frbcesab.r-universe.dev

RemoteUrl https://github.com/FRBCesab/forcis

RemoteRef HEAD

RemoteSha 21fd304122d129c6e6dae7ffca70c499b0a600bb

Contents

`compute_abundances` *Compute count conversions*

Description

Functions to convert species counts between different formats: raw abundance, relative abundance, and number concentration, using counts metadata.

Usage

```
compute_abundances(data, aggregate = TRUE)

compute_concentrations(data, aggregate = TRUE)

compute_frequencies(data, aggregate = TRUE)
```

Arguments

`data` a `data.frame`. One obtained by `read_*_data()` functions.

`aggregate` a `logical` of length 1. If `FALSE` counts will be derived for each subsample. If `TRUE` (default) subsample counts will be aggregated by `sample_id`.

Details

- `compute_concentrations()` converts all counts to number concentrations (n specimens/m³).
- `compute_frequencies()` converts all counts to relative abundances (% specimens per sampling unit).
- `compute_abundances()` converts all counts to raw abundances (n specimens/sampling unit).

Value

A `data.frame`.

`data_to_sf` *Convert a data frame into an sf object*

Description

This function can be used to convert a `data.frame` into an `sf` object. Note that coordinates (columns `site_lon_start_decimal` and `site_lat_start_decimal`) are projected in the Robinson coordinate system.

Usage

```
data_to_sf(data)
```

Arguments

`data` a `data.frame`, i.e. a FORCIS dataset or the output of a `filter_*()` function.

Value

An `sf POINTS` object.

Examples

```
# Attach packages ----
library("forcis")
library("ggplot2")

# Import example dataset ----
file_name <- system.file(file.path("extdata", "FORCIS_net_sample.csv"),
                         package = "forcis")
```

```
net_data <- read.table(file_name, dec = ".", sep = ";")

# Add 'data_type' column ----
net_data$"data_type" <- "Net"

# Dimensions of the data.frame ----
dim(net_data)

# Filter by years ----
net_data_sub <- filter_by_year(net_data, years = 1992)

# Convert to an sf object ----
net_data_sub_sf <- data_to_sf(net_data_sub)

# World basemap ----
ggplot() +
  geom_basemap() +
  geom_sf(data = net_data_sub_sf)
```

download_forcis_db *Download the FORCIS database*

Description

Downloads the entire FORCIS database as a collection of five csv files from Zenodo (https://zenodo.org/doi/10.5281/zenodo.7390791). Additional files will be also downloaded.

Usage

```
download_forcis_db(
  path = ".",
  version = options()$forcis_version,
  check_for_update = options()$check_for_update,
  overwrite = FALSE,
  timeout = 60
)
```

Arguments

path	a character of length 1. The folder in which the FORCIS database will be saved. Note that a subdirectory will be created, e.g. forcis-db/version-99/ (with 99 the version number).
version	a character of length 1. The version number (with two numbers, e.g. 08 instead of 8) of the FORCIS database to use. Default is the latest version. Note that this argument can be handle with the global option forcis_version. For example, if user calls options(forcis_version = "07"), the version 07 will be used by default for the current R session. Note that it is recommended to use the latest version of the database.

check_for_update
: a logical. If TRUE (default) the function will check if a newer version of the FORCIS database is available on Zenodo and will print an informative message. Note that this argument can be handle with the global option check_for_update. For example, if user calls options(check_for_update = FALSE), the message to download the latest version will be disabled for the current R session.

overwrite
: a logical. If TRUE it will override the downloaded files of the FORCIS database. Default is FALSE.

timeout
: an integer. The timeout for downloading files from Zenodo. Default is 60. This number can be increased for low Internet connection.

Details

The FORCIS database is regularly updated. The global structure of the tables doesn't change between versions but some bugs can be fixed and new records can be added. This is why it is recommended to use the latest version of the database. The package is designed to handle the versioning of the database on Zenodo and will inform the user if a new version is available each time he/she uses one of the read_*_data() functions.

For more information, please read the vignette available at https://frbcesab.github.io/forcis/articles/database-versions.html.

Value

No return value. The FORCIS files will be saved in the path folder.

References

Chaabane S, De Garidel-Thoron T, Giraud X, *et al.* (2023) The FORCIS database: A global census of planktonic Foraminifera from ocean waters. *Scientific Data*, 10, 354. DOI: https://doi.org/10.1038/s41597-023-02264-2.

See Also

read_plankton_nets_data() to import the FORCIS database.

Examples

```
## Not run:
# Attach the package ----
library("forcis")

# Folder in which the database will be saved ----
path_to_save_db <- "data"

# Download the database ----
download_forcis_db(path = path_to_save_db)

# Check the content of the folder ----
list.files(path_to_save_db, recursive = TRUE)
```

```
## End(Not run)
```

filter_by_bbox *Filter FORCIS data by a spatial bounding box*

Description

Filters FORCIS data by a spatial bounding box.

Usage

```
filter_by_bbox(data, bbox)
```

Arguments

`data` a `data.frame`. One obtained by `read_*_data()` functions.

`bbox` an object of class `bbox` (package `sf`) or a vector of four `numeric` values defining a square bounding box. Values must follow this order: minimum longitude (`xmin`), minimum latitude (`ymin`), maximum longitude (`xmax`), and maximum latitude (`ymax`). **Important:** if a vector of numeric values is provided, coordinates must be defined in the system WGS 84 (`epsg=4326`).

Value

A `data.frame` containing a subset of `data` for the desired bounding box.

Examples

```
# Attach the package ----
library("forcis")

# Import example dataset ----
file_name <- system.file(file.path("extdata", "FORCIS_net_sample.csv"),
                         package = "forcis")

net_data <- read.table(file_name, dec = ".", sep = ";")

# Add 'data_type' column ----
net_data$"data_type" <- "Net"

# Dimensions of the data.frame ----
dim(net_data)

# Filter by oceans ----
net_data_sub <- filter_by_bbox(net_data, bbox = c(45, -61, 82, -24))

# Dimensions of the data.frame ----
dim(net_data_sub)
```

filter_by_month *Filter FORCIS data by month of sampling*

Description

Filters FORCIS data by month of sampling.

Usage

```
filter_by_month(data, months)
```

Arguments

data	a data.frame. One obtained by read_*_data() functions.
months	a numeric containing one or several months.

Value

A data.frame containing a subset of data for the desired months.

Examples

```
# Attach the package ----
library("forcis")

# Import example dataset ----
file_name <- system.file(file.path("extdata", "FORCIS_net_sample.csv"),
                         package = "forcis")

net_data <- read.table(file_name, dec = ".", sep = ";")

# Add 'data_type' column ----
net_data$"data_type" <- "Net"

# Dimensions of the data.frame ----
dim(net_data)

# Filter by months ----
net_data_sub <- filter_by_month(net_data, months = 1:2)

# Dimensions of the data.frame ----
dim(net_data_sub)
```

filter_by_ocean *Filter FORCIS data by ocean*

Description

Filters FORCIS data by one or several oceans.

Usage

```
filter_by_ocean(data, ocean)
```

Arguments

data	a data.frame. One obtained by read_*_data() functions.
ocean	a character vector of one or several ocean names. Use the function get_ocean_names() to find the correct spelling.

Value

A data.frame containing a subset of data for the desired oceans.

Examples

```
# Attach the package ----
library("forcis")

# Import example dataset ----
file_name <- system.file(file.path("extdata", "FORCIS_net_sample.csv"),
                         package = "forcis")

net_data <- read.table(file_name, dec = ".", sep = ";")

# Add 'data_type' column ----
net_data$"data_type" <- "Net"

# Dimensions of the data.frame ----
dim(net_data)

# Get ocean names ----
get_ocean_names()

# Filter by oceans ----
net_data_sub <- filter_by_ocean(net_data, ocean = "Indian Ocean")

# Dimensions of the data.frame ----
dim(net_data_sub)
```

filter_by_polygon *Filter FORCIS data by a spatial polygon*

Description

Filters FORCIS data by a spatial polygon.

Usage

```
filter_by_polygon(data, polygon)
```

Arguments

data	a data.frame. One obtained by read_*_data() functions.
polygon	an sf POLYGON object.

Value

A data.frame containing a subset of data for the desired spatial polygon.

Examples

```
# Attach the package ----
library("forcis")

# Import example dataset ----
file_name <- system.file(file.path("extdata", "FORCIS_net_sample.csv"),
                         package = "forcis")

net_data <- read.table(file_name, dec = ".", sep = ";")

# Add 'data_type' column ----
net_data$"data_type" <- "Net"

# Dimensions of the data.frame ----
dim(net_data)

# Import Indian Ocean spatial polygons ----
file_name <- system.file(file.path("extdata", "IHO_Indian_ocean_polygon.gpkg"),
                         package = "forcis")

indian_ocean <- sf::st_read(file_name)

# Filter by polygon ----
net_data_sub <- filter_by_polygon(net_data, polygon = indian_ocean)

# Dimensions of the data.frame ----
dim(net_data_sub)
```

filter_by_species *Filter FORCIS data by species*

Description

Filters FORCIS data by a species list.

Usage

```
filter_by_species(data, species)
```

Arguments

data	a `data.frame`. One obtained by `read_*_data()` functions.
species	a `character` vector listing species of interest.

Value

A `data.frame` containing a subset of `data`.

Examples

```
# Import example dataset ----
file_name <- system.file(file.path("extdata", "FORCIS_net_sample.csv"),
                         package = "forcis")

net_data <- read.table(file_name, dec = ".", sep = ";")

# Add 'data_type' column ----
net_data$"data_type" <- "Net"

# Select a taxonomy ----
net_data <- select_taxonomy(net_data, taxonomy = "VT")

# Select only required columns (and taxa) ----
net_data <- select_columns(net_data)

# Dimensions of the data.frame ----
dim(net_data)

# Get species names ----
get_species_names(net_data)

# Select records for three species ----
net_data_sub <- filter_by_species(data    = net_data,
                                  species = c("g_inflata_VT",
                                              "g_elongatus_VT",
                                              "g_glutinata_VT"))
```

```
# Dimensions of the data.frame ----
dim(net_data_sub)

# Get species names ----
get_species_names(net_data_sub)
```

filter_by_year *Filter FORCIS data by year of sampling*

Description

Filters FORCIS data by year of sampling.

Usage

```
filter_by_year(data, years)
```

Arguments

`data`	a `data.frame`. One obtained by `read_*_data()` functions.
`years`	a `numeric` containing one or several years.

Value

A `data.frame` containing a subset of `data` for the desired years.

Examples

```
# Attach the package ----
library("forcis")

# Import example dataset ----
file_name <- system.file(file.path("extdata", "FORCIS_net_sample.csv"),
                         package = "forcis")

net_data <- read.table(file_name, dec = ".", sep = ";")

# Add 'data_type' column ----
net_data$"data_type" <- "Net"

# Dimensions of the data.frame ----
dim(net_data)

# Filter by years ----
net_data_sub <- filter_by_year(net_data, years = 1992)

# Dimensions of the data.frame ----
dim(net_data_sub)
```

geom_basemap *Add a World basemap to a ggplot object*

Description

Creates a World base map that can be added to a ggplot object. Spatial layers come from the Natural Earth project (https://www.naturalearthdata.com/) and are defined in the Robinson coordinate system.

Usage

```
geom_basemap()
```

Value

A ggplot object.

Examples

```
# Attach packages ----
library("forcis")
library("ggplot2")

# World basemap ----
ggplot() +
  geom_basemap()
```

get_available_versions *Get available versions of the FORCIS database*

Description

Gets all available versions of the FORCIS database by querying the Zenodo API (https://developers.zenodo.org).

Usage

```
get_available_versions()
```

Value

A data.frame with three columns:

- publication_date: the date of the release of the version
- version: the label of the version
- access_right: is the version open or restricted?

Examples

```
# Attach the package ----
library("forcis")

# Versions of the FORCIS database ----
get_available_versions()
```

---

get_current_version   *Get the version of the FORCIS database currently used*

Description

Returns the version of the FORCIS database currently used in the project. This function will read the content of the hidden file `.forcis` created by the function `download_forcis_db()`. This file keeps track of the latest version of the database used for a dedicated project. For more information, please read the vignette available at https://frbcesab.github.io/forcis/articles/database-versions.html.

Usage

```
get_current_version()
```

Value

A `character` of length 1, i.e. the label of the version in use.

Examples

```
## Not run:
# Attach the package ----
library("forcis")

# Folder in which the database will be saved ----
path_to_save_db <- "data"

# Download the database ----
download_forcis_db(path = path_to_save_db, version = NULL)

# Get the version of the database ----
get_current_version()

## End(Not run)
```

get_ocean_names *Get World ocean names*

Description

This function returns the name of World oceans according to the IHO Sea Areas dataset version 3 (Flanders Marine Institute, 2018).

Usage

```
get_ocean_names()
```

Value

A `character` vector with World ocean names.

References

Flanders Marine Institute (2018). IHO Sea Areas, version 3. Available online at: https://www.marineregions.org/. DOI: https://doi.org/10.14284/323.

Examples

```
## Not run:
get_ocean_names()

## End(Not run)
```

get_required_columns *Get required column names*

Description

Gets required column names (except taxa names) for the package. This function is designed to help users to add additional columns in `select_columns()` (argument `cols`) if missing from this list.

Usage

```
get_required_columns()
```

Value

A `character`.

Examples

```
# Get required column names (expect taxa names) ----
get_required_columns()
```

get_species_names *Get species names from column names*

Description

Gets species names from column names. This function is just an utility to easily retrieve taxon names.

Usage

```
get_species_names(data)
```

Arguments

data	a data.frame. One obtained by read_*_data() functions.

Value

A data.frame.

Examples

```
## Not run:
# Folder in which the database is stored ----
path_to_db <- "data"

# Download and read the plankton nets data ----
plankton_nets_data <- read_plankton_nets_data(path_to_db)

# Select a taxonomy ----
plankton_nets_data <- select_taxonomy(plankton_nets_data, taxonomy = "OT")

# Retrieve taxon names ----
get_species_names(nets)

## End(Not run)
```

get_version_info *Print information of a specific version of the FORCIS database*

Description

Prints information of a specific version of the FORCIS database by querying the Zenodo API (https://developers.zenodo.org).

Usage

```
get_version_info(version = NULL)
```

Arguments

version — a character of length 1. The label of the version. Use get_available_versions() to list available versions. If NULL (default) the latest version is used.

Value

A list with all information about the version, including: title, doi, publication_date, description, access_right, creators, keywords, version, resource_type, license, and files.

Examples

```
# Attach the package ----
library("forcis")

# Get information for the latest version of the FORCIS database ----
get_version_info()
```

ggmap_data *Map the spatial distribution of FORCIS data*

Description

Maps the spatial distribution of FORCIS data.

Usage

```
ggmap_data(data, col = "red", ...)
```

Arguments

data — a data.frame. One obtained by read_*_data() functions.

col — a character of length 1. The color of data on the map.

... — other graphical parameters passed on to geom_sf().

Value

A ggplot object.

Examples

```
# Attach the package ----
library("forcis")

# Import example dataset ----
file_name <- system.file(file.path("extdata", "FORCIS_net_sample.csv"),
                         package = "forcis")

net_data <- read.table(file_name, dec = ".", sep = ";")
```

```
# Add 'data_type' column ----
net_data$"data_type" <- "Net"

# Map data (default) ----
ggmap_data(net_data)

# Map data ----
ggmap_data(net_data, col = "black", fill = "red", shape = 21, size = 2)
```

`plot_record_by_depth` *Plot sample records by depth of collection*

Description

This function produces a barplot of FORCIS sample records by depth.

Usage

```
plot_record_by_depth(data)
```

Arguments

`data`	a `data.frame`, i.e. a FORCIS dataset.

Value

A `ggplot` object.

Examples

```
# Attach the package ----
library("forcis")

# Import example dataset ----
file_name <- system.file(file.path("extdata", "FORCIS_net_sample.csv"),
                         package = "forcis")

net_data <- read.table(file_name, dec = ".", sep = ";")

# Add 'data_type' column ----
net_data$"data_type" <- "Net"

# Plot data by year (example dataset) ----
plot_record_by_depth(net_data)
```

plot_record_by_month *Plot sample records by month*

Description

This function produces a barplot of FORCIS sample records by month.

Usage

```
plot_record_by_month(data)
```

Arguments

data	a data.frame, i.e. a FORCIS dataset.

Value

A ggplot object.

Examples

```
# Attach the package ----
library("forcis")

# Import example dataset ----
file_name <- system.file(file.path("extdata", "FORCIS_net_sample.csv"),
                         package = "forcis")

net_data <- read.table(file_name, dec = ".", sep = ";")

# Add 'data_type' column ----
net_data$"data_type" <- "Net"

# Plot data by year (example dataset) ----
plot_record_by_month(net_data)
```

plot_record_by_season *Plot sample records by season*

Description

This function produces a barplot of FORCIS sample records by season.

Usage

```
plot_record_by_season(data)
```

Arguments

data	a data.frame, i.e. a FORCIS dataset.

Value

A ggplot object.

Examples

```
# Attach the package ----
library("forcis")

# Import example dataset ----
file_name <- system.file(file.path("extdata", "FORCIS_net_sample.csv"),
                         package = "forcis")

net_data <- read.table(file_name, dec = ".", sep = ";")

# Add 'data_type' column ----
net_data$"data_type" <- "Net"

# Plot data by year (example dataset) ----
plot_record_by_season(net_data)
```

plot_record_by_year *Plot sample records by year*

Description

This function produces a barplot of FORCIS sample records by year.

Usage

```
plot_record_by_year(data)
```

Arguments

data	a data.frame, i.e. a FORCIS dataset.

Value

A ggplot object.

Examples

```
# Attach the package ----
library("forcis")

# Import example dataset ----
file_name <- system.file(file.path("extdata", "FORCIS_net_sample.csv"),
                         package = "forcis")

net_data <- read.table(file_name, dec = ".", sep = ";")

# Add 'data_type' column ----
net_data$"data_type" <- "Net"

# Plot data by year (example dataset) ----
plot_record_by_year(net_data)
```

read_cpr_north_data *Read FORCIS data*

Description

These functions read one specific csv file of the FORCIS database (see below) stored in the folder path. The function download_forcis_db() must be used first to store locally the database.

Usage

```
read_cpr_north_data(
  path = ".",
  version = options()$forcis_version,
  check_for_update = options()$check_for_update
)

read_cpr_south_data(
  path = ".",
  version = options()$forcis_version,
  check_for_update = options()$check_for_update
)

read_plankton_nets_data(
  path = ".",
  version = options()$forcis_version,
  check_for_update = options()$check_for_update
)

read_pump_data(
  path = ".",
  version = options()$forcis_version,
  check_for_update = options()$check_for_update
)

read_sediment_trap_data(
  path = ".",
  version = options()$forcis_version,
  check_for_update = options()$check_for_update
)
```

Arguments

`path` a character of length 1. The folder in which the FORCIS database has been saved.

`version` a character of length 1. The version number (with two numbers, e.g. `08` instead of `8`) of the FORCIS database to use. Default is the latest version. Note that this argument can be handle with the global option `forcis_version`. For example, if user calls `options(forcis_version = "07")`, the version `07` will be used by default for the current R session. Note that it is recommended to use the latest version of the database.

`check_for_update` a `logical`. If `TRUE` (default) the function will check if a newer version of the FORCIS database is available on Zenodo and will print an informative message. Note that this argument can be handle with the global option `check_for_update`. For example, if user calls `options(check_for_update = FALSE)`, the message to download the latest version will be disabled for the current R session.

Details

- `read_plankton_nets_data()` reads the FORCIS plankton nets data
- `read_pump_data()` reads the FORCIS pump data
- `read_cpr_north_data()` reads the FORCIS CPR North data
- `read_cpr_south_data()` reads the FORCIS CPR South data
- `read_sediment_trap_data()` reads the FORCIS sediment traps data

Value

A `data.frame`. See https://zenodo.org/doi/10.5281/zenodo.7390791 for a preview of the datasets.

See Also

`download_forcis_db()` to download the complete FORCIS database.

Examples

```
## Not run:
# Attach the package ----
library("forcis")
```

```
# Folder in which the database will be saved ----
path_to_save_db <- "data"

# Download the database ----
download_forcis_db(path = path_to_save_db)

# Import plankton nets data ----
plankton_nets_data <- read_plankton_nets_data(path = path_to_save_db)

## End(Not run)
```

reshape_data *Reshape and simplify FORCIS data*

Description

Reshapes FORCIS data by pivoting species columns into two columns: `taxa` (taxon names) and `counts` (taxon abundances). It converts wider `data.frame` to a long format.

Usage

```
reshape_data(data)
```

Arguments

`data`	a `data.frame`, i.e. a FORCIS dataset, except for CPR North data.

Value

A `data.frame` reshaped in a long format.

Examples

```
# Attach the package ----
library("forcis")

# Import example dataset ----
file_name <- system.file(file.path("extdata", "FORCIS_net_sample.csv"),
                         package = "forcis")

net_data <- read.table(file_name, dec = ".", sep = ";")

# Add 'data_type' column ----
net_data$"data_type" <- "Net"

# Dimensions of the data.frame ----
dim(net_data)

# Reshape data ----
net_data <- reshape_data(net_data)
```

```
# Dimensions of the data.frame ----
dim(net_data)

# Column names ----
colnames(net_data)
```

`select_columns` *Select columns in FORCIS data*

Description

Selects columns in FORCIS data. Because FORCIS data contains more than 100 columns, this function can be used to lighten the `data.frame` to easily handle it and to speed up some computations.

Usage

```
select_columns(data, cols = NULL)
```

Arguments

`data`	a `data.frame`. One obtained by `read_*_data()` functions.
`cols`	a `character` vector of column names to keep in addition to the required ones (see `get_required_columns()`) and to the taxa columns. Can be `NULL` (default).

Value

A `data.frame`.

Examples

```
# Attach the package ----
library("forcis")

# Import example dataset ----
file_name <- system.file(file.path("extdata", "FORCIS_net_sample.csv"),
                         package = "forcis")

net_data <- read.table(file_name, dec = ".", sep = ";")

# Add 'data_type' column ----
net_data$"data_type" <- "Net"

# Dimensions of the data.frame ----
dim(net_data)

# Select a taxonomy ----
```

```
net_data <- select_taxonomy(net_data, taxonomy = "VT")

# Dimensions of the data.frame ----
dim(net_data)

# Select only required columns (and taxa) ----
net_data <- select_columns(net_data)

# Dimensions of the data.frame ----
dim(net_data)
```

select_taxonomy *Select a taxonomy in FORCIS data*

Description

Selects a taxonomy in FORCIS data. FORCIS database provides three different taxonomies: "LT" (lumped taxonomy), "VT" (validated taxonomy) and "OT" (original taxonomy). See https://doi.org/10.1038/s41597-023-02264-2 for further information.

Usage

```
select_taxonomy(data, taxonomy)
```

Arguments

data	a data.frame. One obtained by read_*_data() functions.
taxonomy	a character of length 1. One among "LT", "VT", "OT".

Value

A data.frame.

Examples

```
# Attach the package ----
library("forcis")

# Import example dataset ----
file_name <- system.file(file.path("extdata", "FORCIS_net_sample.csv"),
                         package = "forcis")

net_data <- read.table(file_name, dec = ".", sep = ";")

# Add 'data_type' column ----
net_data$"data_type" <- "Net"

# Dimensions of the data.frame ----
dim(net_data)
```

```
# Select a taxonomy ----
net_data <- select_taxonomy(net_data, taxonomy = "VT")

# Dimensions of the data.frame ----
dim(net_data)
```

Index