Package ‘forcis’

May 23, 2025

Type Package

Title Handle the FORCIS Foraminifera Database

Version 1.0.1

Description Provides an interface to the 'FORCIS' database (Chaabane et al. (2024) <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://docs.ropensci.org/forcis/`,
`https://github.com/ropensci/forcis`

BugReports `https://github.com/ropensci/forcis/issues`

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.2

VignetteBuilder knitr

Depends R (>= 4.1.0)

Imports ggplot2, httr2, rlang, sf, tibble, tidyr, utils, vroom

Suggests fs, dplyr, httptest2, knitr, rmarkdown, testthat (>= 3.0.0), vdiffr, withr

Config/testthat/edition 3

NeedsCompilation no

Author Nicolas Casajus [aut, cre, cph] (ORCID: <`https://orcid.org/0000-0002-5537-5294`>),
Mattia Greco [aut] (ORCID: <`https://orcid.org/0000-0003-2416-6235`>),
Sonia Chaabane [aut] (ORCID: <`https://orcid.org/0000-0002-4653-8610`>),
Xavier Giraud [aut] (ORCID: <`https://orcid.org/0000-0001-5067-8176`>),
Thibault de Garidel-Thoron [aut] (ORCID: <`https://orcid.org/0000-0001-8983-9571`>),
Khalil Hammami [ctb],
Air Forbes [rev] (ORCID: <`https://orcid.org/0000-0002-9842-7648`>),
FRB-CESAB [fnd]

Maintainer Nicolas Casajus <`nicolas.casajus@fondationbiodiversite.fr`>

Repository CRAN

Date/Publication 2025-05-23 12:02:02 UTC

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 `tibble` or 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 `tibble` in long format with two additional columns: `taxa`, the taxon name and `counts_*`, the number concentration (`counts_n_conc`) or the relative abundance (`counts_rel_ab`) or the raw abundance (`counts_raw_ab`).

Examples

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

net_data <- read.csv(file_name)

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

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

# Compute concentration ----
net_data_conc <- compute_concentrations(net_data)

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

`convert_to_long_format`

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

```
convert_to_long_format(data)
```

Arguments

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

Value

A `tibble` reshaped in a long format.

Examples

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

net_data <- read.csv(file_name)

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

# Reshape data ----
net_data <- convert_to_long_format(net_data)

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

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

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

Value

An `sf POINTS` object.

Examples

```
# Attach package ----
library("ggplot2")

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

net_data <- read.csv(file_name)

# 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()$forcis_version,
  check_for_update = options()$forcis_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 `forcis_check_for_update`. For example, if user calls `options(forcis_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://docs.ropensci.org/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: doi:10.1038/s41597-023022642.

See Also

`read_plankton_nets_data()` to import the FORCIS database.

Examples

```
# Folder in which the database will be saved ----
# N.B. In this example we use a temporary folder but you should select an
# existing folder (for instance "data/").
path <- tempdir()

# Download the database ----
download_forcis_db(path, timeout = 300)

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

`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 `tibble` or 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 `tibble` containing a subset of `data` for the desired bounding box.

Examples

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

net_data <- read.csv(file_name)

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

Value

A `tibble` containing a subset of `data` for the desired months.

Examples

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

net_data <- read.csv(file_name)

# 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 `tibble` or 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 `tibble` containing a subset of `data` for the desired oceans.

Examples

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

net_data <- read.csv(file_name)

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

Value

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

Examples

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

net_data <- read.csv(file_name)

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

Value

A `tibble` 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.csv(file_name)

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

# Select only required columns (and taxa) ----
net_data <- select_forcis_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 `tibble` or a `data.frame`. One obtained by `read_*_data()` functions.
`years`	a `numeric` containing one or several years.

Value

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

Examples

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

net_data <- read.csv(file_name)

# 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 package ----
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 `tibble` 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

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

`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: doi:10.14284/323.

Examples

```
# Print the name of World oceans ----
get_ocean_names()
```

`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_forcis_columns()` (argument `cols`) if missing from this list.

These columns are required by some functions (`compute_*()`, `plot_*()`, etc.) of the package and shouldn't be deleted.

Usage

```
get_required_columns()
```

Value

A `character` vector.

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

Value

A `character` vector of species names.

Examples

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

net_data <- read.csv(file_name)

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

# Retrieve taxon names ----
get_species_names(net_data)
```

`get_version_metadata` *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_metadata(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

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

`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

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

net_data <- read.csv(file_name)

# 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 `tibble` or a `data.frame`, i.e. a FORCIS dataset.

Value

A `ggplot` object.

Examples

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

net_data <- read.csv(file_name)

# 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 `tibble` or a `data.frame`, i.e. a FORCIS dataset.

Value

A `ggplot` object.

Examples

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

net_data <- read.csv(file_name)

# 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 `tibble` or a `data.frame`, i.e. a FORCIS dataset.

Value

A `ggplot` object.

Examples

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

net_data <- read.csv(file_name)

# 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 `tibble` or a `data.frame`, i.e. a FORCIS dataset.

Value

A `ggplot` object.

Examples

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

net_data <- read.csv(file_name)

# 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()$forcis_version,
  check_for_update = options()$forcis_check_for_update
)

read_cpr_south_data(
  path,
  version = options()$forcis_version,
  check_for_update = options()$forcis_check_for_update
)

read_plankton_nets_data(
  path,
  version = options()$forcis_version,
  check_for_update = options()$forcis_check_for_update
)

read_pump_data(
  path,
  version = options()$forcis_version,
  check_for_update = options()$forcis_check_for_update
)

read_sediment_trap_data(
  path,
  version = options()$forcis_version,
  check_for_update = options()$forcis_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 `forcis_check_for_update`. For example, if user calls `options(forcis_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 `tibble`. 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

```
# Folder in which the database will be saved ----
# N.B. In this example we use a temporary folder but you should select an
# existing folder (for instance "data/").
path <- tempdir()

# Download the database ----
download_forcis_db(path, timeout = 300)

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

`select_forcis_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_forcis_columns(data, cols = NULL)
```

Arguments

`data`	a `tibble` or 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 `tibble`.

Examples

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

net_data <- read.csv(file_name)

# 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_forcis_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 doi:10.1038/s41597023022642 for further information.

Usage

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

Arguments

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

Value

A `tibble`.

Examples

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

net_data <- read.csv(file_name)

# 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