Package: funbiogeo (via r-universe)

August 24, 2024

Type Package

Title Functional Biogeography Analyses

Version 0.0.0.9000

Description Helps users with analyses in functional biogeogeography by loading and combining data, computing trait coverage, as well as computing functional diversity indices, drawing maps, correlating them with the environment, and upscaling assemblages.

URL `https://frbcesab.github.io/funbiogeo/`, `https://github.com/frbcesab/funbiogeo`

BugReports `https://github.com/frbcesab/funbiogeo/issues`

License GPL (>= 2)

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports ggplot2 (>= 3.4.0), rlang, scales, sf, stats, terra, tidyr, tidyselect, utils, xfun

Suggests ggridges, knitr, mockery, patchwork, rmarkdown, rstudioapi, testthat (>= 3.0.0)

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

VignetteBuilder knitr

Config/testthat/edition 3

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

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

RemoteRef HEAD

RemoteSha 3c7d5d1c6e58c9b51648d65533cd0f9f586f3763

Contents

`fb_aggregate_site_data` *Aggregate Site Data Along Coarser Spatial Grid*

Description

This function helps aggregating site data along a coarser grid. The idea is that you have any type of data at the site scale (diversity metrics, environmental data, etc.) but you would like to work on it at a coarser scale, or you want to visualize it at that scale. This function helps you do exactly that.

Usage

```
fb_aggregate_site_data(site_locations, site_data, agg_grid, fun = mean)
```

Arguments

`site_locations`	an `sf` object with the spatial geometries of sites. **NOTE**: the first column should be named `"site"` and indicate site names.
`site_data`	a `matrix` or `data.frame` containing values per sites to aggregate along the provided grid. Can have one or several columns (variables to aggregate). The first column must contain sites names as provided in the example dataset `site_locations`.
`agg_grid`	a `SpatRaster` object (package `terra`). A raster of one single layer, that defines the grid along which to aggregate.
`fun`	the function used to aggregate points values when there are multiple points in one cell. Default is `mean`.

Value

A `SpatRaster` object with as many layers as columns in `site_data`.

Examples

```
library("funbiogeo")

data("site_locations")
data("site_species")

## Import grid ----
tavg <- system.file("extdata", "annual_mean_temp.tif", package = "funbiogeo")
tavg <- terra::rast(tavg)

## Rasterize 3 first species counts ----
fb_aggregate_site_data(site_locations, site_species[, 1:4], tavg, fun = sum)
```

`fb_count_sites_by_species` *Count Number of Sites Occupied by Species*

Description

For each species computes the percentage of sites where the species is present (distribution value higher than 0 and non-NA).

Usage

```
fb_count_sites_by_species(site_species)
```

Arguments

`site_species`	a `data.frame` with sites in rows and species in columns. **NOTE**: the first column should be named `"site"` and indicate site names. The other columns should be named according to species names.

Value

A three-column `data.frame` with:

- `species`: the name of the species;
- `n_sites`: the number of sites where the species is present;
- `coverage`: the percentage of sites where the species is present.

Examples

```
library("funbiogeo")

data("site_species")

site_coverage_by_species <- fb_count_sites_by_species(site_species)
head(site_coverage_by_species)
```

`fb_count_species_by_site` *Count Number of Species per Site*

Description

For each site computes the proportion of species present (distribution value higher than 0 and non-NA) compared to all species provided. For example, a site could contain only 20% of all species provided.

Usage

```
fb_count_species_by_site(site_species)
```

Arguments

`site_species`	a `data.frame` with sites in rows and species in columns. **NOTE**: the first column should be named `"site"` and indicate site names. The other columns should be named according to species names.

Value

A three-column `data.frame` with:

- `site`: the name of the site;
- `n_species`: the number of present species;
- `coverage`: the percentage of present species.

Examples

```
library("funbiogeo")

data("site_species")

species_coverage_by_site <- fb_count_species_by_site(site_species)
head(species_coverage_by_site)
```

`fb_count_species_by_trait`

Count Number of Species for Each Trait

Description

For each trait computes the percentage of species without `NA` (missing trait values).

Usage

```
fb_count_species_by_trait(species_traits)
```

Arguments

`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.

Value

A three-column `data.frame` with:

- `trait`: the name of the trait;
- `n_species`: the number of species with non-missing value for the trait;
- `coverage`: the percentage of species with non-missing value for the trait.

Examples

```
library("funbiogeo")

data("species_traits")

species_coverage_by_trait <- fb_count_species_by_trait(species_traits)
head(species_coverage_by_trait)
```

`fb_count_traits_by_species` *Compute Number of Known Trait(s) per Species*

Description

For each species computes the percentage of traits without `NA` (missing trait values).

Usage

```
fb_count_traits_by_species(species_traits)
```

Arguments

`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.

Value

A three-column `data.frame` with:

- `species`: the name of the species;
- `n_traits`: the number of traits with non-missing value for the species;
- `coverage`: the percentage of traits with non-missing value for the species.

Examples

```
library("funbiogeo")

data("species_traits")

trait_coverage_by_species <- fb_count_traits_by_species(species_traits)
head(trait_coverage_by_species)
```

`fb_cwm` *Compute community-weighted means (CWM) of trait values*

Description

This function returns the community-weighted mean of provided trait values. It only works with quantitative traits and will warn you otherwise. It will remove species that either have `NA` values in the `site_species` input or `NA` values as their trait.

Usage

```
fb_cwm(site_species, species_traits)
```

Arguments

`site_species`	a `data.frame` with sites in rows and species in columns. **NOTE**: the first column should be named `"site"` and indicate site names. The other columns should be named according to species names.
`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.

Value

A `data.frame` with sites in rows and the following variables:

- `site`, the site label,
- `trait`, the trait label as provided in `species_traits`,
- and `cwm`, the community-weighted means of quantitative traits values.

Examples

```
library("funbiogeo")

data("site_species")
data("species_traits")

site_cwm <- fb_cwm(head(site_species), species_traits)
head(site_cwm)
```

`fb_filter_sites_by_species_coverage` *Filter sites with a given species coverage threshold*

Description

Selects sites (rows) for which the percentage of present species (distribution value higher than 0 and non-NA) is higher than a threshold.

Usage

```
fb_filter_sites_by_species_coverage(
  site_species,
  threshold_species_proportion = 0
)
```

Arguments

`site_species`	a `data.frame` with sites in rows and species in columns. **NOTE**: the first column should be named `"site"` and indicate site names. The other columns should be named according to species names.
`threshold_species_proportion`	a numeric of length 1 between 0 and 1. The percentage of species coverage threshold.

Value

A subset of `site_species` with sites covered by X% of species.

Examples

```
library("funbiogeo")

data("site_species")

# Get sites with more than 40% of the species
new_site_species <- fb_filter_sites_by_species_coverage(
  site_species,
  threshold_species_proportion = 0.4
)
# There are now only 148 sites
dim(new_site_species)
new_site_species[1:3, 1:4]
```

`fb_filter_sites_by_trait_coverage`

Filter sites with a given trait coverage threshold

Description

...

Usage

```
fb_filter_sites_by_trait_coverage(
  site_species,
  species_traits,
  threshold_traits_proportion = 1
)
```

Arguments

`site_species`	a `data.frame` with sites in rows and species in columns. **NOTE**: the first column should be named `"site"` and indicate site names. The other columns should be named according to species names.
`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.
`threshold_traits_proportion`	a numeric of length 1 between 0 and 1. The percentage trait coverage threshold

Value

A subset of `site_species` with sites covered by X% of abundance/coverage considering all provided traits.

Examples

```
library("funbiogeo")

data("site_species")
data("species_traits")

# Filter all the sites where all species have known traits
new_site_species <- fb_filter_sites_by_trait_coverage(
  site_species, species_traits
)

# There is only one such site
nrow(new_site_species)

# Filter sites where at least 80% of species have known traits
```

```
new_site_species_2 <- fb_filter_sites_by_trait_coverage(
  site_species, species_traits, threshold_traits_proportion = 0.8
)

# There are now four sites
nrow(new_site_species_2)
```

`fb_filter_species_by_site_coverage`

Filter species with a given sites coverage threshold

Description

Selects species (columns) for which the percentage of sites where the species is present (distribution value higher than 0 and non-NA) is higher than a threshold.

Usage

```
fb_filter_species_by_site_coverage(
  site_species,
  threshold_sites_proportion = 0
)
```

Arguments

`site_species`	a `data.frame` with sites in rows and species in columns. **NOTE**: the first column should be named `"site"` and indicate site names. The other columns should be named according to species names.
`threshold_sites_proportion`	a numeric of length 1 between 0 and 1. The percentage of sites coverage threshold.

Value

A subset of `site_species` with species with a prevalence higher than `threshold_sites_proportion`.

Examples

```
library("funbiogeo")

data("site_species")

# Filter species present in at least 10% of the sites
new_site_species <- fb_filter_species_by_site_coverage(
  site_species,
  threshold_sites_proportion = 0.1
)

new_site_species[1:3, 1:4]
```

```
# There are now only 84 species (instead of 149)
ncol(new_site_species)
```

`fb_filter_species_by_trait_coverage`

Filter species with a given traits coverage threshold

Description

Selects species (rows) for which the percentage of traits without `NA` (missing trait values) is higher than a threshold.

Usage

```
fb_filter_species_by_trait_coverage(
  species_traits,
  threshold_traits_proportion = 0
)
```

Arguments

`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.
`threshold_traits_proportion`	a numeric of length 1 between 0 and 1. The percentage of traits coverage threshold.

Value

A subset of `species_traits` with species covered by X% of traits.

Examples

```
library("funbiogeo")

data("species_traits")

# Filter species that have at least 60% of the traits described
new_species_traits <- fb_filter_species_by_trait_coverage(
  species_traits,
  threshold_traits_proportion = 0.6
)

# There are now only 93 species
nrow(new_species_traits)
```

`fb_filter_traits_by_species_coverage` *Filter traits with a given species coverage threshold*

Description

Selects traits (columns) for which the percentage of species without `NA` (missing trait values) is higher than a threshold.

Usage

```
fb_filter_traits_by_species_coverage(
  species_traits,
  threshold_species_proportion = 0
)
```

Arguments

`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.
`threshold_species_proportion`	`numeric(1)` [default = `NULL`] between 0 and 1. The percentage of species coverage threshold.

Value

A subset of `species_traits` with traits for the specified proportion of species.

Examples

```
library("funbiogeo")

data("species_traits")

# Filter traits that have at least 60% non-missing values
new_species_traits <- fb_filter_traits_by_species_coverage(
  species_traits,
  threshold_species_proportion = 0.6
)

# There are now only 2 traits
ncol(new_species_traits)
```

`fb_format_site_locations` *Extract site x locations information from long format data*

Description

Convert a flat `data.frame` with site coordinates into a proper `sf` object that can then be used by other functions. This function assumes that the coordinates are given in WGS84 (longitude vs. latitude). The function automatically removes repeated coordinates from the input dataset.

Usage

```
fb_format_site_locations(
  data,
  site,
  longitude,
  latitude,
  crs = sf::st_crs(4326),
  na_rm = FALSE
)
```

Arguments

`data`	a `data.frame` in a long format (see example).
`site`	a `character` of length 1. Name of the column with site labels.
`longitude`	a `character` of length 1. Name of the column with longitude. The function assumes coordinates are WGS84 (EPSG:4326).
`latitude`	a `character` of length 1. Name of the column with latitude. The function assumes coordinates are WGS84 (EPSG:4326).
`crs`	a `character` of length 1 or an object of class `crs`. Coordinate Reference System (CRS) of the specified coordinates. The CRS should be a valid CRS in R. It can either be a character like `"+proj=longlat +datum=WGS84 +no_defs"` or as specified using `sf::st_crs()` like `sf::st_crs(4326)` the default value.
`na_rm`	a logical value. If `TRUE` remove sites with incomplete coordinates. Default is `FALSE`.

Value

An `sf` object with a `site` column specifying site coordinates.

Examples

```
library("funbiogeo")

filename <- system.file("extdata", "raw_mammals_data.csv",
                        package = "funbiogeo")
```

```
all_data <- read.csv(filename)

head(all_data)

site_locations <- fb_format_site_locations(all_data, "site", "longitude",
                                           "latitude")
head(site_locations)
```

`fb_format_site_species`

Extract site x species information from long format data

Description

Convert a flat `data.frame` with species occurrence/abundance at site level into a proper `data.frame` object that can then be used by other functions. The final output contains sites in rows and species in columns.

Usage

```
fb_format_site_species(data, site, species, value, na_to_zero = TRUE)
```

Arguments

`data`	a `data.frame` in a long format (see example).
`site`	a `character` of length 1. Name of the column with site labels.
`species`	a `character` of length 1. Name of the column with species names.
`value`	a `character` of length 1. Name of the column with species occurrence/abundance.
`na_to_zero`	a logical value. If `TRUE` (default) `NA` are replaced by `0`.

Value

A `data.frame` with sites in rows and species in columns. The first column is named `"site"` and contains the name of the sites.

Examples

```
library("funbiogeo")

filename <- system.file("extdata", "raw_mammals_data.csv",
                        package = "funbiogeo")
all_data <- read.csv(filename)

head(all_data)

site_species <- fb_format_site_species(all_data, "site", "species", "count")
site_species[1:3, 1:4]
```

`fb_format_species_categories` *Extract species x categories information from long format data*

Description

Convert a flat `data.frame` with species names and species (supra-)category (e.g. family, order, endemism status, etc.) into a proper `data.frame` object that can then be used by other functions. The final output contains species in rows and two columns (species name and species category).

Usage

```
fb_format_species_categories(data, species, category)
```

Arguments

`data`	a `data.frame` in a long format (see example).
`species`	a `character` of length 1. Name of the column with species names.
`category`	a `character` of length 1. Name of the column with species category.

Value

A `data.frame` with species in rows and two columns (species name and species category).

Examples

```
library("funbiogeo")

filename <- system.file("extdata", "raw_mammals_data.csv",
                        package = "funbiogeo")
all_data <- read.csv(filename)

head(all_data)

species_categories <- fb_format_species_categories(all_data, "species",
                                                   "order")
species_categories[1:6, ]
```

---

fb_format_species_traits

Extract species x traits information from long format data

Description

Convert a flat `data.frame` with traits values for different species into a proper `data.frame` object that can then be used by other functions. The final output contains species in rows and traits in columns.

Usage

```
fb_format_species_traits(data, species, traits)
```

Arguments

`data`	a `data.frame` in a long format (see example).
`species`	a `character` of length 1. Name of the column with species names.
`traits`	a `character` of length >= 1. Name(s) of trait column(s).

Value

A `data.frame` with species in rows and traits in columns, with the first column names `"species"` containing the species names.

Examples

```
library("funbiogeo")

filename <- system.file("extdata", "raw_mammals_data.csv",
                        package = "funbiogeo")
all_data <- read.csv(filename)

head(all_data)

traits <- c("adult_body_mass", "gestation_length", "litter_size",
            "max_longevity", "sexual_maturity_age", "diet_breadth")

species_traits <- fb_format_species_traits(all_data, "species", traits)
head(species_traits)
```

`fb_get_all_trait_coverages_by_site` *Compute Trait Coverage per Site for Each Trait*

Description

Compute trait coverage for all sites, i.e., the percentage of total abundance/presence of species that have traits data compared to total species. This function assumes that all species provided in the traits dataset have all their traits specified (meaning that all species have either known or `NA` values reported as their traits). The coverage of each trait separately is returned as well as all traits taken together if wanted.

Usage

```
fb_get_all_trait_coverages_by_site(
  site_species,
  species_traits,
  all_traits = TRUE
)
```

Arguments

`site_species`	a `data.frame` with sites in rows and species in columns. **NOTE**: the first column should be named `"site"` and indicate site names. The other columns should be named according to species names.
`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.
`all_traits`	a logical (default = `TRUE`) which tell if the coverage considering all provided traits should be provided in an additional column `all_traits`

Value

a data.frame with a column with sites and one column per provided trait giving its coverage (percent species per site, weighted by abundance that have trait data), and, when argument `all_traits = TRUE`, an additional column named `all_traits` considering the coverage of all traits taken together.

Examples

```
site_trait_cov <- fb_get_all_trait_coverages_by_site(
  site_species, species_traits
)

head(site_trait_cov)
```

`fb_get_environment` *Extract Raster Values at Location of Sites*

Description

Extract Raster Values at Location of Sites

Usage

```
fb_get_environment(site_locations, environment_raster)
```

Arguments

`site_locations`	an `sf` object with the spatial geometries of sites. **NOTE**: the first column should be named `"site"` and indicate site names.
`environment_raster`	a `SpatRaster` object (package `terra`). A single or multi-layers environmental raster.

Value

A `data.frame` with average environmental values (columns) per site (rows), with the first column being `"site"` indicating site names.

Examples

```
library("funbiogeo")

data("site_locations")

## Import climate rasters ----
prec <- system.file("extdata", "annual_tot_prec.tif", package = "funbiogeo")
tavg <- system.file("extdata", "annual_mean_temp.tif", package = "funbiogeo")

layers <- terra::rast(c(tavg, prec))

fb_get_environment(head(site_locations), layers)
```

`fb_get_trait_combination_coverage` *Compute site trait coverage for each trait combination*

Description

This function computes trait coverage for each site for different trait combinations. If not provided, consider all possible trait combinations. The function will not run if the total number of combinations given is over 10,000.

Usage

```
fb_get_trait_combination_coverage(
  site_species,
  species_traits,
  comb_size = NULL
)
```

Arguments

`site_species`	a `data.frame` with sites in rows and species in columns. **NOTE**: the first column should be named `"site"` and indicate site names. The other columns should be named according to species names.
`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.
`comb_size`	an integer vector defining one or more sizes of combinations (default: `NULL`)

Value

a data.frame with the following columns:

- `site` with the site indices from `site_species`,
- `combination_length` with the number of traits in given combinations,
- `combination_name` with the name of the trait combination (concatenated trait names with `__`),
- `trait_coverage` the corresponding trait coverage for the given trait combination and site.

Examples

```
# Compute Coverages using All Trait Combinations
all_combinations = fb_get_trait_combination_coverage(
site_species, species_traits
)

# Get only combinations of 3 traits
three_traits = fb_get_trait_combination_coverage(
site_species, species_traits, 3
)

# Combinations of 2, 3, and 4 traits
two_to_four = fb_get_trait_combination_coverage(
site_species, species_traits, c(2, 3, 4)
)
```

`fb_get_trait_coverage_by_site` *Compute Trait Coverage For Each Site Weighted by Abundance*

Description

Compute trait coverage for all sites, i.e., the percentage of total abundance/presence of species that have traits data compared to total species. This function assumes that all species provided in the traits dataset have all their traits specified (meaning that all species have either known or `NA` values reported as their traits). **NB**: this function returns trait coverage using all traits provided in the input `species_traits` data.frame.

Usage

```
fb_get_trait_coverage_by_site(site_species, species_traits)
```

Arguments

`site_species`	a `data.frame` with sites in rows and species in columns. **NOTE**: the first column should be named `"site"` and indicate site names. The other columns should be named according to species names.
`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.

Value

A `data.frame` with `n` rows (where `n` is the number of sites) and two columns: `site`, the site label, and `trait_coverage`, the percent of total abundance/presence of species that have traits data.

Examples

```
library("funbiogeo")

data("site_species")
data("species_traits")

site_trait_cov <- fb_get_trait_coverage_by_site(site_species, species_traits)

head(site_trait_cov)
```

`fb_make_report` *Create an R Markdown Report on Site/Species/Trait Coverage with Plots*

Description

Creates an R Markdown (`.Rmd`) report from a template to explore and summarize users data. User can modify this report and use the function `rmarkdown::render()` to convert this `.Rmd` in different formats:

- HTML document (`output_format = "bookdown::html_document2"`);
- PDF document (`output_format = "bookdown::pdf_document2"`);
- Word document (`output_format = "bookdown::word_document2"`);
- HTML, PDF and Word documents (`output_format = "all"`).

Usage

```
fb_make_report(
  path = ".",
  filename = NULL,
  title = NULL,
  author = NULL,
  species_traits_name,
  site_species_name,
  site_locations_name,
  overwrite = FALSE,
  open = TRUE
)
```

Arguments

`path`	a `character` of length 1. The directory in which the `.Rmd` file will be created. This directory must exist.
`filename`	a `character` of length 1. The name of the `.Rmd` file to be created. If `NULL` (default) the `.Rmd` file will be named `funbiogeo_report.Rmd`.
`title`	a `character` of length 1. The title of the report. If `NULL` (default) the title will be `funbiogeo Report`.
`author`	a `character` of length 1. The author(s) of the report. If `NULL` (default) no author will be added.
`species_traits_name`	a `character` of length 1. The **name** of the species x traits dataset (not the object). Note that before rendering the report this dataset must be loaded.
`site_species_name`	a `character` of length 1. The **name** of the sites x species dataset (not the object). Note that before rendering the report this dataset must be loaded.

`site_locations_name`	a `character` of length 1. The **name** of the sites x locations dataset (not the object). Note that before rendering the report this dataset must be loaded.
`overwrite`	a logical. If this file is already present and `overwrite = TRUE`, it will be erased and replaced by the template. Default is `FALSE`.
`open`	a logical. If `TRUE` (default), this file will be opened on the text editor.

Value

No return value.

Examples

```
# Create temporary folder
temp_path <- tempdir()

# Load data
data("species_traits")
data("site_species")
data("site_locations")

# Create report
fb_make_report(
  path                = temp_path,
  author              = "Casajus N. and Grenié M.",
  species_traits_name = "species_traits",
  site_species_name   = "site_species",
  site_locations_name = "site_locations",
  open = FALSE
)

## Not run:
# Open Rmd file ----
utils::file.edit(file.path(temp_path, "funbiogeo_report.Rmd"))

# Render Rmd file ----
rmarkdown::render(file.path(temp_path, "funbiogeo_report.Rmd"),
                  output_format = "all")

## End(Not run)
```

`fb_map_raster` *Map a Single Raster Layer*

Description

Map a Single Raster Layer

Usage

```
fb_map_raster(x, ...)
```

Arguments

`x`	a `SpatRaster` object (package `terra`). A raster of one single layer
`...`	other parameters passed to `theme()`

Value

A `ggplot` object.

Examples

```
library(ggplot2)

## Load raster ----
tavg <- system.file("extdata", "annual_mean_temp.tif", package = "funbiogeo")
tavg <- terra::rast(tavg)

## Default map ----
fb_map_raster(tavg)

## Map with custom theme ----
fb_map_raster(tavg, legend.position = "bottom")

## Advanced customization ----
my_map <- fb_map_raster(tavg) +
  scale_fill_distiller("Temperature", palette = "Spectral") +
  theme(legend.position = "bottom") +
  ggtitle("Mean annual temperature in Western Europe")

my_map

## Map w/o annotation ----
fb_map_raster(tavg) +
  theme_void() +
  theme(legend.position = "none")
```

`fb_map_site_data` *Map Arbitrary Site Data*

Description

From the site-locations data and a dataset organized by site, plot a map of this information. The returned plot is as little customized as possible to let the user do the customization.

Usage

```
fb_map_site_data(site_locations, site_data, selected_col)
```

Arguments

`site_locations`	an `sf` object with the spatial geometries of sites. **NOTE**: the first column should be named `"site"` and indicate site names.
`site_data`	`data.frame()` of additional site information containing the column `"site"` to merge with the `site_locations` argument
`selected_col`	`character(1)` name of the column to plot

Value

a `ggplot` object.

Examples

```
site_rich <- fb_count_species_by_site(site_species)

# Map of Species Richness
rich_map <- fb_map_site_data(site_locations, site_rich, "n_species")
rich_map

# Customize the map
rich_map +
  ggplot2::scale_fill_viridis_c("Species Richness")
```

`fb_map_site_traits_completeness`
Map Trait Coverage Per Site

Description

Returns a `ggplot2` map of sites colored by trait coverage (proportion of species having a known trait value). By default shows one plot for each trait and add an additional facet named `"all_traits"` considering the trait coverage with all traits taken together.

Usage

```
fb_map_site_traits_completeness(
  site_locations,
  site_species,
  species_traits,
  all_traits = TRUE
)
```

Arguments

`site_locations`	an `sf` object with the spatial geometries of sites. **NOTE**: the first column should be named `"site"` and indicate site names.
`site_species`	a `data.frame` with sites in rows and species in columns. **NOTE**: the first column should be named `"site"` and indicate site names. The other columns should be named according to species names.
`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.
`all_traits`	a logical (default = `TRUE`) which tell if the coverage considering all provided traits should be provided in an additional column `all_traits`

Value

a 'ggplot2' object

Examples

```
fb_map_site_traits_completeness(site_locations, site_species, species_traits)
```

`fb_plot_distribution_site_trait_coverage`

Plot Distribution of Trait Coverages across all Sites

Description

Plots the distributions of trait coverage across site, i.e. the proportion of species weighted by abundance with known trait values, for each trait separately and all traits taken together. The trait distributions are ordered from the lowest to the highest average trait coverage (top to bottom). The top always displays a distribution named `"all_traits"` that contains the distribution of coverage all traits taken together.

Usage

```
fb_plot_distribution_site_trait_coverage(
  site_species,
  species_traits,
  species_categories = NULL,
  all_traits = TRUE
)
```

Arguments

`site_species`	a `data.frame` with sites in rows and species in columns. **NOTE**: the first column should be named `"site"` and indicate site names. The other columns should be named according to species names.
`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.
`species_categories`	(default = `NULL`) 2-columns `data.frame` giving species categories, with the first column describing the species name, and the second column giving their corresponding categories
`all_traits`	a logical (default = `TRUE`) which tell if the coverage considering all provided traits should be provided in an additional column `all_traits`

Value

a 'ggplot2' object

Examples

```
fb_plot_distribution_site_trait_coverage(site_species, species_traits)
```

`fb_plot_number_sites_by_species`

Plot Number of Sites by Species

Description

Represent all species in each function of the number of sites they occupy. The species are ordered from the ones that occupy the least number of sites from the ones that occupy the most. The number of site is indicated at the bottom x-axis, while the top x-axis represents the proportion of occupied sites. The left y-axis label species names and their rank by increasing prevalence. The user can supplied a threshold of sites to see how many species occupy more or less than the given proportion of sites.

Usage

```
fb_plot_number_sites_by_species(
  site_species,
  threshold_sites_proportion = NULL
)
```

Arguments

`site_species`	a `data.frame` with sites in rows and species in columns. **NOTE**: the first column should be named `"site"` and indicate site names. The other columns should be named according to species names.
`threshold_sites_proportion`	a numeric of length 1 between 0 and 1. The percentage of sites coverage threshold.

Value

a `ggplot2` object

Examples

```
fb_plot_number_sites_by_species(site_species)

# Add a vertical cutoff line (40% of sites)
fb_plot_number_sites_by_species(site_species, 0.4)
```

`fb_plot_number_species_by_trait`

Plot Number of Species per Trait

Description

Display a lollipop graph showing the number and proportion of species with non-NA trait for each trait ranked in decreasing order.

Usage

```
fb_plot_number_species_by_trait(
  species_traits,
  species_categories = NULL,
  threshold_species_proportion = NULL
)
```

Arguments

`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.
`species_categories`	(default = `NULL`) 2-columns `data.frame` giving species categories, with the first column describing the species name, and the second column giving their corresponding categories
`threshold_species_proportion`	`numeric(1)` [default = `NULL`] between 0 and 1. The percentage of species coverage threshold.

Value

a ggplot2 object

Examples

```
data(species_traits)

fb_plot_number_species_by_trait(species_traits)

# Add a vertical cutoff line (12.5% of species)
fb_plot_number_species_by_trait(species_traits, NULL, 1/8)
```

`fb_plot_number_traits_by_species` *Plot Number of Traits per Species*

Description

Display a graph showing the number (and proportion) of species having at least 0, 1, etc. number of traits. It provides a lollipop graph to examine which number of trait cover a certain proportion of the species. This plot doesn't show which traits are concerned.

Usage

```
fb_plot_number_traits_by_species(
  species_traits,
  species_categories = NULL,
  threshold_species_proportion = NULL
)
```

Arguments

`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.
`species_categories`	(default = `NULL`) 2-columns `data.frame` giving species categories, with the first column describing the species name, and the second column giving their corresponding categories
`threshold_species_proportion`	`numeric(1)` [default = `NULL`] between 0 and 1. The percentage of species coverage threshold.

Value

a `ggplot2` object

Examples

```
data(species_traits)

fb_plot_number_traits_by_species(species_traits)

# Add a vertical cutoff line (33% of the species)
fb_plot_number_traits_by_species(
  species_traits, threshold_species_proportion = 1/3
 )
```

`fb_plot_site_environment`

Plot Position of Sites in Environmental Space

Description

Plot a figure showing the average environmental space of given sites compared to a full environmental vector. For the sake of simplicity only represents the figure along two environmental axes. The average environmental value are extracted for each site.

Usage

```
fb_plot_site_environment(
  site_locations,
  environment_raster,
  first_layer = names(environment_raster)[1],
  second_layer = names(environment_raster)[2]
)
```

Arguments

`site_locations`	an `sf` object with the spatial geometries of sites. **NOTE**: the first column should be named `"site"` and indicate site names.
`environment_raster`	a `SpatRaster` object (package `terra`). A single or multi-layers environmental raster.
`first_layer`	`character(1)` the name of the first layer to use, by default uses the first layer of `environment_raster`
`second_layer`	`character(1)` the name of the second layer to use, by default uses the second layer of `environment_raster`

Value

a `ggplot` object

Examples

```
data("site_locations")

# Import climate rasters
prec <- system.file("extdata", "annual_tot_prec.tif", package = "funbiogeo")
tavg <- system.file("extdata", "annual_mean_temp.tif", package = "funbiogeo")

layers <- terra::rast(c(tavg, prec))

# Make plot (show environmental position of 6 first sites)
fb_plot_site_environment(head(site_locations), layers)
```

`fb_plot_site_traits_completeness`
Plot Trait Coverage per Site for each Trait

Description

Display a binary heatmap visualizing the site x traits matrix with colors displaying the proportion of occurring species with known trait values. Traits are ordered from the most to the least known (left to right). Sites are ordered from the ones with highest to lowest overall trait coverage (bottom to top). The site average proportion of species with known trait for each trait (across all sites) is shown in the x-axis labels. An additional column at the very right of the plot named `"all_traits"` shows a summary considering traits together.

Usage

```
fb_plot_site_traits_completeness(
  site_species,
  species_traits,
  species_categories = NULL,
  all_traits = TRUE
)
```

Arguments

`site_species`	a `data.frame` with sites in rows and species in columns. **NOTE**: the first column should be named `"site"` and indicate site names. The other columns should be named according to species names.
`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.
`species_categories`	(default = `NULL`) 2-columns `data.frame` giving species categories, with the first column describing the species name, and the second column giving their corresponding categories
`all_traits`	a logical (default = `TRUE`) which tell if the coverage considering all provided traits should be provided in an additional column `all_traits`

Value

a ggplot2 object

Examples

```
fb_plot_site_traits_completeness(site_species, species_traits)
```

`fb_plot_species_traits_completeness`

Plot Trait Coverage per Species for each Trait

Description

Display a binary heatmap visualizing the species x traits matrix with colors displaying present and missing traits. Traits are ordered from the most to the least known (left to right). Species are ordered from the ones with most to the ones with least traits (bottom to top). The proportion of species with non-missing traits is shown on the x-axis labels. An additional column at the very right of the plot named `"all_traits"` shows a summary considering if all other traits are known.

Usage

```
fb_plot_species_traits_completeness(
  species_traits,
  species_categories = NULL,
  all_traits = TRUE
)
```

Arguments

`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.
`species_categories`	(default = `NULL`) 2-columns `data.frame` giving species categories, with the first column describing the species name, and the second column giving their corresponding categories
`all_traits`	a logical (default = `TRUE`) which tell if the coverage considering all provided traits should be provided in an additional column `all_traits`

Value

a `ggplot2` object

Examples

```
data(species_traits)
fb_plot_species_traits_completeness(species_traits)
```

`fb_plot_species_traits_missingness` *Plot Trait Coverage per Species for each Trait*

Description

Display a binary heatmap visualizing the species x traits matrix with colors displaying present and missing traits. Traits are ordered from the most to the least known (left to right). Species are ordered from the ones with most to the ones with least traits (bottom to top). The proportion of species with non-missing traits is shown on the x-axis labels. An additional column at the very right of the plot named `"all_traits"` shows a summary considering if all other traits are known.

Usage

```
fb_plot_species_traits_missingness(
  species_traits,
  species_categories = NULL,
  all_traits = TRUE
)
```

Arguments

`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.
`species_categories`	2-columns `data.frame` giving species categories `NULL` by default, with the first column describing the species name, and the second column giving their corresponding categories
`all_traits`	a logical (default = `TRUE`) which tell if the coverage considering all provided traits should be provided in an additional column `all_traits`

Value

a `ggplot2` object

Examples

```
data(species_traits)
fb_plot_species_traits_missingness(species_traits)
```

`fb_plot_trait_combination_frequencies` *Plot Present and Missing Trait Combinations Frequencies*

Description

Display a figure of Present/Missing Trait Combinations where one square is represents one trait. The y-axis gives the frequency of the row (as well as its proportions compared to the total number of rows).

Usage

```
fb_plot_trait_combination_frequencies(
  species_traits,
  species_categories = NULL,
  order_by = c("number", "complete")
)
```

Arguments

`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.
`species_categories`	(default = `NULL`) 2-columns `data.frame` giving species categories, with the first column describing the species name, and the second column giving their corresponding categories
`order_by`	`character(1)` either `"number"` or `"complete` If `"number"` order rows by frequency so that most frequent rows are at the bottom. Otherwise order rows to order table by the number of non-missing traits then by the frequency of combinations

Value

a `ggplot2` object

Examples

```
# Ordered by number by default
fb_plot_trait_combination_frequencies(species_traits)

# Order by present traits
fb_plot_trait_combination_frequencies(species_traits, NULL, "complete")
```

`fb_plot_trait_correlation` *Plot Trait Correlation Matrix*

Description

This functions outputs a `ggplot2` figure that outputs a trait correlation matrix. It only works on numerical traits and will output messages if non-numerical traits are found and errors if no numerical traits are found. Internally it uses the `stats::cor()` function and only works on complete observation (it removes any row that contains any `NA`). Use the `...` argument to pass options to the `cor()` function.

Usage

```
fb_plot_trait_correlation(species_traits, species_categories = NULL, ...)
```

Arguments

`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.
`species_categories`	(default = `NULL`) 2-columns `data.frame` giving species categories, with the first column describing the species name, and the second column giving their corresponding categories
`...`	Additional options passed to `stats::cor()`

Value

a `ggplot` object

Examples

```
fb_plot_trait_correlation(species_traits)

# Plot Spearman's correlation
fb_plot_trait_correlation(species_traits, method = "spearman")
```

`fb_table_trait_summary`

Summary Table on Traits (Missingness, Range, etc.)

Description

This function outputs a `data.frame` that summarises the species by trait table to have many information in a glance. This can then return a data.frame or a nicely formatted `knitr::kable()` for inclusion in an Rmarkdown document.

Usage

```
fb_table_trait_summary(species_traits, kable = FALSE)
```

Arguments

`species_traits`	a `data.frame` with species in rows and traits as columns. **NOTE**: The first column should be named `"species"` and contain species names. The other columns should be named according to trait names.
`kable`	`TRUE` or `FALSE` Should function returns a `knitr::kable()`? defaults `FALSE`

Value

a `data.frame` with the following columns:

- `trait_name`: a `character` column with the trait name as indicated in `species_traits`
- `trait_type`: the nature of the trait (`numeric`, `categorical`, or `ordered`)
- `number_non_missing`: the total number of non-`NA` trait values
- `proportion_non_missing`: the proportion of non-`NA` trait values
- `trait_range`: for numerical traits, the range of values
- `trait_mean_sd`: for numerical traits, the mean plus-minus the standard deviation
- `number_distinct`: for non-numerical traits, the number of categories
- `list_distinct`: for non-numerical traits, the list of categories

Examples

```
# Get a data.frame back
fb_table_trait_summary(species_traits)

# Get a kable (to use in Rmd documents)
fb_table_trait_summary(species_traits, TRUE)
```

`site_locations` *Sites x Locations of Western Europe Mammals*

Description

This dataset contains spatial coordinates (grid cells of 0.5° x 0.5° horizontal resolution) of sites of Western Europe where mammals species presence/absence (see `site_species` dataset) has been sampled.

Usage

```
site_locations
```

Format

An `sf` object with 1,505 rows (grid cells) with a WGS84 (EPSG:4236) Coordinate Reference System.

Details

See `https://github.com/frbcesab/eumammals` for a complete description of the workflow used to create this dataset.

This dataset shows the format of the argument `site_locations` used in several functions of `funbiogeo`.

The variable `site` corresponds to the sites labels.

Examples

```
data("site_locations")
class(site_locations)
dim(site_locations)
head(site_locations)
```

`site_species` *Sites x Species of Western Europe Mammals*

Description

This dataset contains the presence/absence of mammals species in Western Europe. This dataset is derived from IUCN range maps downloaded at: `https://www.iucnredlist.org/resources/spatial-data-download`. In respect to the license, species names have been anonymized.

Usage

```
site_species
```

Format

A `data.frame` with 1,505 rows (sites) and 150 columns (1 column for site label and 149 for species occurrence).

Details

See https://github.com/frbcesab/eumammals for a complete description of the workflow used to create this dataset.

This dataset shows the format of the argument `site_species` used in several functions of `funbiogeo`.

Note that sites labels (`site`) is the first column of this dataset.

References

IUCN (2021) The IUCN Red List of Threatened Species. Version 2021-2. Downloaded on 2021/10/11.

Examples

```
data("site_species")
class(site_species)
dim(site_species)
site_species[1:6, 1:6]
```

`species_categories` *Species x Categories of Western Europe Mammals*

Description

This dataset contains the order of mammals species in Western Europe. This dataset is derived from IUCN range maps downloaded at: https://www.iucnredlist.org/resources/spatial-data-download. In respect to the license, species names have been anonymized.

Usage

```
species_categories
```

Format

A `data.frame` with 149 rows (species) and 2 columns (1 column for species name and 1 for the order).

Details

See https://github.com/frbcesab/eumammals for a complete description of the workflow used to create this dataset.

This dataset shows the format of the argument `species_categories` used in several functions of `funbiogeo`.

Note that species names (`species`) is the first column of this dataset.

References

IUCN (2021) The IUCN Red List of Threatened Species. Version 2021-2. Downloaded on 2021/10/11.

Examples

```
data("species_categories")
class(species_categories)
dim(species_categories)
species_categories[1:6, ]
```

`species_traits` *Species x Traits of Western Europe Mammals*

Description

This dataset contains values for six functional traits of Western Europe mammals species. These values were extracted from the **PanTHERIA** database (Jones 2009), a database on World mammals traits available at: `https://esapubs.org/archive/ecol/E090/184/metadata.htm`. In respect to the IUCN license, species names have been anonymized.

Usage

```
species_traits
```

Format

A matrix with 149 rows (species) and the following seven columns:

species species name corresponding to the columns of `site_species`

adult_body_mass adult body mass (in grams)

gestation_length length of time of non-inactive fetal growth (in days)

litter_size number of offspring born per litter per female

max_longevity maximum adult age (in months)

sexual_maturity_age age when individuals are first physically capable of reproducing (in days)

diet_breadth number of dietary categories eaten

Details

See `https://github.com/frbcesab/eumammals` for a complete description of the workflow used to create this dataset.

This dataset shows the format of the argument `species_traits` used in several functions of `funbiogeo`.

Note that species names (`species`) is the first column of this dataset.

References

Jones KE *et al.* (2009) PanTHERIA: A species-level database of life history, ecology, and geography of extant and recently extinct mammals. *Ecology*, **90**, 2648. DOI: 10.1890/08-1494.1.

Examples

```
data("species_traits")
class(species_traits)
dim(species_traits)
head(species_traits)
```

Index