Package ‘florabr’

July 22, 2025

Title Explore Flora e Funga do Brasil Database

Version 1.3.0

Description A collection of functions designed to retrieve, filter and spatialize data from the Flora e Funga do Brasil dataset. For more information about the dataset, please visit <https://floradobrasil.jbrj.gov.br/consulta/>.

Imports XML (>= 3.99.0.14), data.table (>= 1.14.8), httr (>= 1.4.6), terra (>= 1.7.39), stats (>= 4.2.3), utils(>= 4.2.3), grDevices (>= 4.2.3), doSNOW (>= 1.0.20), parallel (>= 4.3.1), foreach (>= 1.5.2)

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Depends R (>= 2.10)

LazyData true

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

URL https://wevertonbio.github.io/florabr/

BugReports https://github.com/wevertonbio/florabr/issues

Config/testthat/edition 3

NeedsCompilation no

Author Weverton Trindade [aut, cre] (ORCID: <https://orcid.org/0000-0003-2045-4555>)

Maintainer Weverton Trindade <wevertonf1993@gmail.com>

Repository CRAN

Date/Publication 2024-09-09 17:00:02 UTC

Contents

`bf_data` *Flora e Funga do Brasil database - Version 393.401*

Description

A dataset containing a subset of the Flora e Funga do Brasil database (version 393.401)

Usage

```
data(bf_data)
```

Format

A `data.frame` with 50010 rows and 23 variables:

species Species names

scientificName Complete scientific name of the species

acceptedName Accepted name of the species (NA when the name in species is already an accepted name)

kingdom Kingdom to which species belongs (Plantae or Fungi)

group Major group to which species belongs (Angiosperms, Gymnosperms, Ferns and Lycophytes, Bryophytes, and Algae)

subgroup Subgroup to which species belongs. Only available for Bryophytes (Mosses, Hornworts, and Liverworts)

phylum Phylum to which species belongs

class Class to which species belongs

order Order to which species belongs

family Family to which species belongs

genus Genus to which species belongs

lifeForm Life form of the species (e.g: Tree, Herb, Shrub, etc.)

habitat Habitat type of the species (e.g., Terrestrial, Rupicolous, Epiphytic, etc.)

biome Biomes with confirmed occurrences of the species

states Federal states with confirmed occurrences of the species

vegetation Vegetation types with confirmed occurrences of the species

origin Indicates whether the species is Native, Naturalized, or Cultivated in Brazil

endemism Indicates whether the species is Endemic or Non-endemic to Brazil

taxonomicStatus Indicates the level of recognition and acceptance of the species (Accepted or Synonym)

nomenclaturalStatus Indicates the legitimacy and validity of the species name (Correct, Illegitimate, Uncertain_Application, etc.)

vernacularName Locally or culturally used name for the species

taxonRank Taxonomic rank (Species, Genus, Family, Order, etc). This data contains only Species

id Unique id for species

References

Flora e Funga do Brasil. Jardim Botânico do Rio de Janeiro. Available at: http://floradobrasil.jbrj.gov.br/

`biomes` *SpatVector of the biomes of Brazil*

Description

A simplified and packed SpatVector of the polygons of the biomes present in Brazilian territory. The spatial data was originally obtained from `geobr::read_biomes`. Borders have been simplified by removing vertices of borders using `terra::simplifyGeom`. It's necessary unpack the Spatvectos using `terra::unwrap`

@usage data(biomes) biomes <- terra::unwrap(biomes)

Usage

```
biomes
```

Format

A `SpatVector` with 6 geometries and 1 attribute:

name_biome The name of the biome (Amazon, Caatinga, Cerrado, Atlantic_Forest, Pampa, and Pantanal)

`brazil` *SpatVector of the Brazil's national borders*

Description

A simplified and packed SpatVector of the Brazil's national borders. The spatial data was originally obtained from `geobr::read_country`. Borders have been simplified by removing vertices of borders using `terra::simplifyGeom`. It's necessary unpack the Spatvectos using `terra::unwrap`

@usage data(brazil) brazil <- terra::unwrap(brazil)

Usage

```
brazil
```

Format

A `SpatVector` with 1 geometry and 0 attribute

`check_names` *Check species names*

Description

`check_names` checks if the species names are correct and searches for suggestions if the name is misspelled or not found in the Flora e Funga do Brasil database

`match_names` finds approximate matches to the specified pattern (species) within each element of the string `x` (species_to_match). It is used internally by `check_names`.

Usage

```
check_names(data, species, max_distance = 0.1,
                   include_subspecies= FALSE, include_variety = FALSE,
                   kingdom = "Plantae", parallel = FALSE, ncores = 1,
                   progress_bar = FALSE)

match_names(
  species,
  species_to_match,
  max_distance = 0.1,
```

```
    parallel = FALSE,
    ncores = 1,
    progress_bar = FALSE
  )
```

Arguments

`data`	(data.frame) the data.frame imported with the `load_florabr` function.
`species`	(character) names of the species to be checked.
`max_distance`	(numeric) Maximum distance (as a fraction) allowed for searching suggestions when the name is misspelled. It can be any value between 0 and 1. The higher the value, the more suggestions are returned. For more details, see `agrep`. Default = 0.1.
`include_subspecies`	(logical) whether to include subspecies. Default = FALSE
`include_variety`	(logical) whether to include varieties. Default = FALSE
`kingdom`	(character) the kingdom to which the species belong. It can be "Plantae" or "Fungi". Default = "Plantae".
`parallel`	(logical) whether to run in parallel. Setting this to `TRUE` is recommended for improved performance when working with 100 or more species.
`ncores`	(numeric) number of cores to use for parallel processing. Default is 1. This is only applicable if `parallel` = `TRUE`.
`progress_bar`	(logical) whether to display a progress bar during processing. Default is FALSE
`species_to_match`	(character) a vector of species names to match against the `species` parameter.

Value

a data.frame with the following columns:

- input_name: the species names informed in species argument
- Spelling: indicates if the species name is Correct (a perfect match with a species name in the Flora e Funga do Brasil), Probably_incorrect (partial match), or Not_found (no match with any species).
- Suggested name: If Spelling is Correct, it is the same as the input_name. If Spelling is Probably_correct, one or more suggested names are listed, found according to the maximum distance. If Spelling is "Not_found", the value is NA.
- Distance: The integer Levenshtein edit distance. It represents the number of single-character edits (insertions, deletions, or substitutions) required to transform the input_name into the Suggested_name.
- taxonomicStatus: the taxonomic status of the species name ("Accepted" or "Synonym").
- nomenclaturalStatus: the nomenclatural status of the species name. This information is not available for all species.
- acceptedName: If the species name is not accepted or incorrect, the accepted name of the specie. If the species name is accepted and correct, the same as input_name and Suggested_name.
- family: the family of the specie.

References

Flora e Funga do Brasil. Jardim Botânico do Rio de Janeiro. Available at: http://floradobrasil.jbrj.gov.br/

Examples

```
data("bf_data", package = "florabr")
spp <- c("Butia cattarinensis", "Araucaria angustifolia")
check_names(data = bf_data, species = spp)
```

`check_version` *Check if you have the latest version of Flora e Funga do Brasil data available*

Description

This function checks if you have the latest version of the Flora e Funga do Brasil data available in a specified directory.

Usage

```
check_version(data_dir)
```

Arguments

`data_dir`	the directory where the data should be located.

Value

A message informing whether you have the latest version of Flora e Funga do Brasil available in the data_dir

Examples

```
#Check if there is a version of Flora e Funga do Brasil data available in the
#current directory
check_version(data_dir = getwd())
```

`filter_florabr` *Identify records outside natural ranges according to Flora e Funga do Brasil*

Description

This function removes or flags records outside of the species' natural ranges according to information provided by the Flora e Funga do Brasil database.

Usage

```
filter_florabr(data, occ, species = "species", long = "x", lat = "y",
                          by_state = TRUE, buffer_state = 20, by_biome = TRUE,
                          buffer_biome = 20, by_endemism = TRUE,
                          buffer_brazil = 20, state_vect = NULL,
                          state_column = NULL, biome_vect = NULL,
                          biome_column = NULL, br_vect = NULL,
                          value = "flag&clean", keep_columns = TRUE,
                          verbose = TRUE)
```

Arguments

`data`	(data.frame) the data.frame imported with the `load_florabr` function.
`occ`	(data.frame) a data.frame with the records of the species.
`species`	(character) column name in occ with species names. Default = "species"
`long`	(character) column name in occ with longitude data. Default = "x"
`lat`	(character) column name in occ with latitude data. Default = "y"
`by_state`	(logical) filter records by state? Default = TRUE
`buffer_state`	(numeric) buffer (in km) around the polygons of the states of occurrence of the specie. Default = 20.
`by_biome`	(logical) filter records by biome? Default = TRUE
`buffer_biome`	(numeric) buffer (in km) around the polygons of the biomes of occurrence of the specie. Default = 20.
`by_endemism`	(logical) filter records by endemism? Default = TRUE
`buffer_brazil`	(numeric) buffer (in km) around the polygons of the brazil. Default = 20.
`state_vect`	(SpatVector) a SpatVector of the Brazilian states. By default, it uses the SpatVector provided by geobr::read_state(). It can be another Spatvector, but the structure must be identical to geobr::read_state().
`state_column`	(character) name of the column in state_vect containing state abbreviations. Only use if biome_vect is not null.
`biome_vect`	(SpatVector) a SpatVector of the Brazilian biomes. By default, it uses the SpatVector provided by geobr::read_biomes(). It can be another SpatVector, but the structure must be identical to geobr::read_biomes() with biome names in English.

`biome_column`	(character) name of the column in biome_vect containing names of brazilian biomes (in English: "Amazon", "Atlantic_Forest", "Caatinga", "Cerrado", "Pampa" and "Pantanal". Only use if biome_vect is not null.
`br_vect`	(SpatVector) a SpatVector of brazil. By default, it uses the SpatVector provided by geobr::read_state() after being aggregated/dissolved,
`value`	(character) Defines output values. See Value section. Default = "flag&clean".
`keep_columns`	(logical) if TRUE, keep all the original columns of the input occ. If False, keep only the columns species, long and lat. Default = TRUE
`verbose`	(logical) Whether to display species being filtered during function execution. Set to TRUE to enable display, or FALSE to run silently. Default = TRUE.

Details

If by_state = TRUE and/or by_biome = TRUE, the function takes polygons representing the states and/or biomes with confirmed occurrences of the specie, draws a buffer around the polygons, and tests if the records of the species fall inside it. If by_endemism = TRUE, the function checks if the species is endemic to brazil. If it is endemic, the function tests if the records of the specie fall inside a polygon representing the boundaries of brazil (with a buffer).

Value

Depending on the 'value' argument. If value = "flag", it returns the same data.frame provided in data with additional columns indicating if the record falls inside the natural range of the specie (TRUE) or outside (FALSE). If value = "clean", it returns a data.frame with only the records that passes all the tests (TRUE for all the filters). If value = "flag&clean" (Default), it returns a list with two data.frames: one with the flagged records and one with the cleaned records.

References

Flora e Funga do Brasil. Jardim Botânico do Rio de Janeiro. Available at: http://floradobrasil.jbrj.gov.br/

Examples

```
data("bf_data") #Load Flora e Funga do Brasil data
data("occurrences") #Load occurrences
pts <- subset(occurrences, species == "Myrcia hatschbachii")
fd <- filter_florabr(data = bf_data, occ = pts,
                    by_state = TRUE, buffer_state = 20,
                    by_biome = TRUE, buffer_biome = 20,
                    by_endemism = TRUE, buffer_brazil = 20,
                    state_vect = NULL,
                    biome_vect = NULL, br_vect = NULL,
                    value = "flag&clean", keep_columns = TRUE,
                    verbose = FALSE)
```

`get_attributes` *Get available attributes to filter species*

Description

This function displays all the options available to filter species by its characteristics

Usage

```
get_attributes(data, attribute)
```

Arguments

`data`	(data.frame) a data.frame imported with the `load_florabr` function or a data.frame generated with the `select_species` function.
`attribute`	(character) the type of characteristic. Accept more than one option. See detail to see the options.

Details

The attribute argument accepts the following options: kingdom, group, subgroup, phylum, class, order, family, lifeform, habitat, vegetation, origin, endemism, biome, states, taxonomicstatus or nomenclaturalstatus. These options represent different characteristics of species that can be used for filtering.

Value

a list of data.frames with the available options to use in the `select_species` function.

References

Flora e Funga do Brasil. Jardim Botânico do Rio de Janeiro. Available at: http://floradobrasil.jbrj.gov.br/

Examples

```
data("bf_data") #Load Flora e Funga do Brasil data
# Get available biomes, life forms and states to filter species
d <- get_attributes(data = bf_data,
                    attribute = c("biome", "lifeform", "states"))
```

`get_binomial` *Extract the binomial name (Genus + specific epithet + infraspecific epithet (optional)) from a full Scientific Name*

Description

Extract the binomial name (Genus + specific epithet + infraspecific epithet (optional)) from a full Scientific Name

Usage

```
get_binomial(species_names,
                     include_subspecies = TRUE,
                     include_variety = TRUE)
```

Arguments

`species_names`	(character) Scientific names to be converted to binomial names
`include_subspecies`	(logical) include subspecies? If TRUE (default), the function extracts any infraspecific epithet after the pattern "subsp."
`include_variety`	(logical) include subspecies? If TRUE (default), the function extracts any infraspecific epithet after the pattern "var."

Value

A vector with the binomial names (Genus + specific epithet).

Examples

```
spp <- c("Araucaria angustifolia (Bertol.) Kuntze",
         "Araucaria angustifolia var. alba Reitz",
         "Butia catarinensis Noblick & Lorenzi",
         "Butia eriospatha subsp. punctata",
         "Adesmia paranensis Burkart")
spp_new <- get_binomial(species_names = spp,
                        include_subspecies = TRUE,
                        include_variety = TRUE)
spp_new
```

`get_florabr` *Download the latest version of Flora e Funga do Brasil database*

Description

This function downloads the latest or an older version of Flora e Funga do Brasil database, merges the information into a single data.frame, and saves this data.frame in the specified directory.

Usage

```
get_florabr(output_dir, data_version = "latest",
                 solve_discrepancy = FALSE, overwrite = TRUE,
                 verbose = TRUE, remove_files = TRUE)
```

Arguments

`output_dir`	(character) a directory to save the data downloaded from Flora e Funga do Brasil.
`data_version`	(character) Version of the Flora e Funga do Brasil database to download. Use "latest" to get the most recent version, updated weekly. Alternatively, specify an older version (e.g., data_version = "393.319"). Default value is "latest".
`solve_discrepancy`	Resolve discrepancies between species and subspecies/varieties information. When set to TRUE, species information is updated based on unique data from varieties and subspecies. For example, if a subspecies occurs in a certain biome, it implies that the species also occurs in that biome. Default = FALSE.
`overwrite`	(logical) If TRUE, data is overwritten. Default = TRUE.
`verbose`	(logical) Whether to display messages during function execution. Set to TRUE to enable display, or FALSE to run silently. Default = TRUE.
`remove_files`	(logical) Whether to remove the downloaded files used in building the final dataset. Default is TRUE.

Value

The function downloads the latest version of the Flora e Funga do Brasil database from the official source. It then merges the information into a single data.frame, containing details on species, taxonomy, occurrence, and other relevant data. The merged data.frame is then saved as a file in the specified output directory. The data is saved in a format that allows easy loading using the `load_florabr` function for further analysis in R.

References

Flora e Funga do Brasil. Jardim Botânico do Rio de Janeiro. Available at: http://floradobrasil.jbrj.gov.br/

Examples

```
## Not run:
#Creating a folder in a temporary directory
#Replace 'file.path(tempdir(), "florabr")' by a path folder to be create in
#your computer
my_dir <- file.path(file.path(tempdir(), "florabr"))
dir.create(my_dir)
#Download, merge and save data
get_florabr(output_dir = my_dir, data_version = "latest",
            solve_discrepancy = FALSE, overwrite = TRUE, verbose = TRUE)

## End(Not run)
```

`get_pam` *Get a presence-absence matrix*

Description

Get a presence-absence matrix of species based on its distribution (states, biomes and vegetation types) according to Flora e Funga do Brasil.

Usage

```
get_pam(data, by_biome = TRUE, by_state = TRUE,
                 by_vegetation = FALSE, remove_empty_sites = TRUE,
                 return_richness_summary = TRUE,
                 return_spatial_richness = TRUE,
                 return_plot = TRUE)
```

Arguments

`data`	(data.frame) a data.frame imported with the `load_florabr` function or generated by either `select_species` or `subset_species` functions
`by_biome`	(logical) get occurrences by biome. Default = TRUE
`by_state`	(logical) get occurrences by State. Default = TRUE
`by_vegetation`	(logical) get occurrences by vegetation type. Default = FALSE
`remove_empty_sites`	(logical) remove empty sites (sites without any species) from final presence-absence matrix. Default = TRUE
`return_richness_summary`	(logical) return a data.frame with the number of species in each site. Default = TRUE
`return_spatial_richness`	(logical) return a SpatVector with the number of species in each site. Default = TRUE
`return_plot`	(logical) plot map with the number of species in each site. Only works if return_spatial_richness = TRUE. Default = TRUE

Value

If return_richness_summary and/or return_spatial_richness is set to TRUE, return a list with:

- PAM: the presence-absence matrix (PAM)
- Richness_summary: a data.frame with the number of species in each site
- Spatial_richness: a SpatVector with the number of species in each site (only by State and biome)

If return_richness_summary and return_spatial_richness is set to FALSE, return a presence-absence matrix

References

Flora e Funga do Brasil. Jardim Botânico do Rio de Janeiro. Available at: http://floradobrasil.jbrj.gov.br/

Examples

```
data("bf_data") #Load Flora e Funga do Brasil data
#Select endemic and native species of trees with occurrence only in Amazon
am_trees <- select_species(data = bf_data,
                          include_subspecies = FALSE,
                          include_variety = FALSE,
                          kingdom = "Plantae",
                          group = "All", subgroup = "All",
                          family = "All", genus = "All",
                          lifeForm = "Tree", filter_lifeForm = "only",
                          habitat = "All", filter_habitat = "in",
                          biome = "Amazon",
                          filter_biome = "only",
                          state = "All", filter_state = "and",
                          vegetation = "All",
                          filter_vegetation = "in",
                          endemism = "Endemic", origin = "Native",
                          taxonomicStatus = "Accepted",
                          nomenclaturalStatus = "All")
#Get presence-absence matrix
pam_am <- get_pam(data = am_trees, by_biome = TRUE, by_state = TRUE,
                 by_vegetation = FALSE, remove_empty_sites = TRUE,
                 return_richness_summary = TRUE,
                 return_spatial_richness = TRUE,
                 return_plot = TRUE)
```

`get_spat_occ` *Get Spatial polygons (SpatVectors) of species based on its distribution (states and biomes) according to Flora e Funga do Brasil*

Description

Get Spatial polygons (SpatVectors) of species based on its distribution (states and biomes) according to Flora e Funga do Brasil

Usage

```
get_spat_occ(
  data,
  species,
  state = TRUE,
  biome = TRUE,
  intersection = TRUE,
  state_vect = NULL,
  state_column = NULL,
  biome_vect = NULL,
  biome_column = NULL,
  verbose = TRUE
)
```

Arguments

`data`	(data.frame) the data.frame imported with the `load_florabr` function.
`species`	(character) one or more species names (only genus and specific epithet, eg. "Araucaria angustifolia")
`state`	(logical) get SpatVector of states with occurrence of the species? Default = TRUE
`biome`	(logical) get SpatVector of biomes with occurrence of the species? Default = TRUE
`intersection`	(character) get a Spatvector representing the intersection between states and biomes with occurrence of the specie? To use intersection = TRUE, you must define state = TRUE and biome = TRUE". Default = TRUE
`state_vect`	(SpatVector) a SpatVector of the Brazilian states. By default, it uses the SpatVector provided by geobr::read_state(). It can be another Spatvector, but the structure must be identical to geobr::read_state().
`state_column`	(character) name of the column in state_vect containing state abbreviations. Only use if biome_vect is not null.
`biome_vect`	(SpatVector) a SpatVector of the Brazilian biomes. By default, it uses the SpatVector provided by geobr::read_biomes(). It can be another SpatVector, but the structure must be identical to geobr::read_biomes().
`biome_column`	(character) name of the column in biome_vect containing names of brazilian biomes (in English: "Amazon", "Atlantic_Forest", "Caatinga", "Cerrado", "Pampa" and "Pantanal". Only use if biome_vect is not null.
`verbose`	(logical) Whether to display species being filtered during function execution. Set to TRUE to enable display, or FALSE to run silently. Default = TRUE.

Value

A list with SpatVectors of states and/or biomes and/or Intersections for each specie.

References

Flora e Funga do Brasil. Jardim Botânico do Rio de Janeiro. Available at: http://floradobrasil.jbrj.gov.br/

Examples

```
library(terra)
data("bf_data") #Load Flora e Funga do Brasil data
spp <- c("Araucaria angustifolia", "Adesmia paranensis") #Example species
#Get states, biomes and intersection states-biomes of species
spp_spt <- get_spat_occ(data = bf_data, species = spp, state = TRUE,
                       biome = TRUE, intersection = TRUE, state_vect = NULL,
                       biome_vect = NULL, verbose = TRUE)


#Plot states of occurrence of Araucaria angustifolia
plot(spp_spt[[1]]$states, main = names(spp_spt)[[1]])
#Plot biomes of occurrence of Araucaria angustifolia
plot(spp_spt[[2]]$biomes, main = names(spp_spt)[[2]])
#Plot intersection between states and biomes of occurrence of
#Araucaria angustifolia
plot(spp_spt[[1]]$states_biomes)
```

`get_synonym` *Retrieve synonyms for species*

Description

Retrieve synonyms for species

Usage

```
get_synonym(data, species,
                     include_subspecies = TRUE, include_variety = TRUE)
```

Arguments

`data`	(data.frame) the data.frame imported with the `load_florabr` function
`species`	(character) names of the species
`include_subspecies`	(logical) include subspecies that are synonyms of the species? Default = TRUE
`include_variety`	(logical) include varieties that are synonyms of the species? Default = TRUE

Value

A data.frame containing unique synonyms of the specified species along with relevant information on taxonomic and nomenclatural statuses.

References

Flora e Funga do Brasil. Jardim Botânico do Rio de Janeiro. Available at: http://floradobrasil.jbrj.gov.br/

Examples

```
data("bf_data") #Load Flora e Funga do Brasil data
#Species to extract synonyms
spp <- c("Araucaria angustifolia", "Adesmia paranensis")
spp_synonyms <- get_synonym(data = bf_data, species = spp,
                            include_subspecies = TRUE,
                            include_variety = TRUE)
spp_synonyms
```

`load_florabr` *Load Flora e Funga do Brasil database*

Description

Load Flora e Funga do Brasil database

Usage

```
load_florabr(data_dir, data_version = "Latest_available",
                    type = "short", verbose = TRUE)
```

Arguments

`data_dir`	(character) the same directory used to save the data downloaded from Flora e Funga do Brasil using the get_florabr function.
`data_version`	(character) the version of Flora e Funga do Brasil database to be loaded. It can be "Latest_available", which will load the latest version available; or another specified version, for example "393.364". Default = "Latest_available".
`type`	(character) it determines the number of columns that will be loaded. It can be "short" or "complete". Default = "short". See details.
`verbose`	(logical) Whether to display messages during function execution. Set to TRUE to enable display, or FALSE to run silently. Default = TRUE.

Details

The parameter type accepts two arguments. If type = short, it will load a data.frame with the 20 columns needed to run the other functions of the package: species, scientificName, acceptedName, kingdom, Group, Subgroup, family, genus, lifeForm, habitat, Biome, States, vegetationType, Origin, Endemism, taxonomicStatus, nomenclaturalStatus, vernacularName, taxonRank, and id If type = complete, it will load a data.frame with all 39 variables available in Flora e Funga do Brasil database.

Value

A data.frame with the specified version (Default is the latest available) of the Flora e Funga do Brasil database. This data.frame is necessary to run most of the functions of the package.

References

Flora e Funga do Brasil. Jardim Botânico do Rio de Janeiro. Available at: http://floradobrasil.jbrj.gov.br/

Examples

```
## Not run:
#Creating a folder in a temporary directory
#Replace 'file.path(tempdir(), "florabr")' by a path folder to be create in
#your computer
my_dir <- file.path(file.path(tempdir(), "florabr"))
dir.create(my_dir)
#Download, merge and save data
get_florabr(output_dir = my_dir, data_version = "latest", overwrite = TRUE,
            verbose = TRUE)
#Load data
df <- load_florabr(data_dir = my_dir, data_version = "Latest_available",
type = "short")

## End(Not run)
```

`occurrences` *Records of plant species*

Description

A dataset containing records of 7 plant species downloaded from GBIF. The records were obtained with `plantR::rgbif2`

Usage

```
data(occurrences)
```

Format

A `data.frame` with 1521 rows and 3 variables:

- **species** Species names (Araucaria angustifolia, Abatia americana, Passiflora edmundoi, Myrcia hatschbachii, Serjania pernambucensis, Inga virescens, and Solanum restingae)
- **x** Longitude
- **y** Latitude

References

GBIF, 2024. florabr R package: Records of plant species. https://doi.org/10.15468/DD.QPGEB7

`select_by_vernacular` *Search for taxa using vernacular names*

Description

Search for taxa using vernacular names

Usage

```
select_by_vernacular(data, names, exact = FALSE)
```

Arguments

`data`	(data.frame) the data.frame imported with the `load_florabr` function or generated with the function `select_species`.
`names`	(character) vernacular name ("Nome comum") of the species to be searched
`exact`	(logic) if TRUE, the function will search only for exact matches. For example, if names = "pinheiro" and exact = TRUE, the function will return only the species popularly known as "pinheiro". On the other hand, if names = "pinheiro" and exact = FALSE, the function will return other results as "pinheiro-do-parana". Default = FALSE

Value

a data.frame with the species with vernacular names that match the input names

References

Flora e Funga do Brasil. Jardim Botânico do Rio de Janeiro. Available at: http://floradobrasil.jbrj.gov.br/

Flora e Funga do Brasil. Jardim Botânico do Rio de Janeiro. Available at: http://floradobrasil.jbrj.gov.br/

Examples

```
data("bf_data") #Load Flora e Funga do Brasil data
#Search for species whose vernacular name is 'pinheiro'
pinheiro_exact <- select_by_vernacular(data = bf_data,
                                       names = "pinheiro",
                                       exact = TRUE)
pinheiro_exact
#Search for species whose vernacular name is 'pinheiro', allowing non-exact
#matches
pinheiro_not_exact <- select_by_vernacular(data = bf_data,
                                          names = "pinheiro",
                                          exact = FALSE)
head(pinheiro_not_exact)
```

`select_species` *Selection of species based on its characteristics and distribution*

Description

select_species allows filter species based on its characteristics and distribution available in Flora e Funga do Brasil

Usage

```
select_species(data,
                        include_subspecies = FALSE, include_variety = FALSE,
                        kingdom = "Plantae", group = "All", subgroup = "All",
                        phylum = "All", class ="All", order = "All",
                        family = "All", genus = "All",
                        lifeForm = "All", filter_lifeForm = "in",
                        habitat = "All", filter_habitat = "in",
                        biome = "All", filter_biome = "in",
                        state = "All", filter_state = "in",
                        vegetation = "All", filter_vegetation = "in",
                        endemism = "All", origin = "All",
                        taxonomicStatus = "Accepted",
                        nomenclaturalStatus = "All")
```

Arguments

`data`	(data.frame) the data.frame imported with the `load_florabr` function.
`include_subspecies`	(logical) include subspecies? Default = FALSE
`include_variety`	(logical) include varieties of the species? Default = FALSE

`kingdom`	(character) The kingdom for filtering the dataset. It can be "Plantae" or "Fungi". Default = "Plantae". To include both, use c("Plantae", "Fungi")
`group`	(character) The groups for filtering the datasets. It can be "Fungi", "Angiosperms", "Gymnosperms", "Ferns and Lycophytes", "Bryophytes" and "Algae". To use more than one group, put the available items in a vector, for example: group = c(Angiosperms", "Gymnosperms"). Default = "All".
`subgroup`	(character) The subgroups for filtering the dataset. Only available if the group is "Fungi" or "Bryophytes". For Fungi, it can be "stricto sensu" or "lato sensu". For Bryophytes, it can be "Mosses", "Hornworts" and "Liverworts" . To use more than one group, put the available items in a vector, for example: subgroup = c("Mosses", "Hornworts"). Default = "All".
`phylum`	(character) The phyla for filtering the dataset. It can be included more than one phylum. Default = "All".
`class`	(character) The classes for filtering the dataset. It can be included more than one class. Default = "All".
`order`	(character) The orders for filtering the dataset. It can be included more than one order. Default = "All".
`family`	(character) The families for filtering the dataset. It can be included more than one family. Default = "All".
`genus`	(character) The genus for filtering the dataset. It can be included more than one genus. Default = "All".
`lifeForm`	(character) The life forms for filtering the dataset. It can be included more than one lifeForm. Default = "All"
`filter_lifeForm`	(character) The type of filtering for life forms. It can be "in", "only", "not_in" and "and". See details for more about this argument.
`habitat`	(character) The life habitat for filtering the dataset. It can be included more than one habitat. Default = "All"
`filter_habitat`	(character) The type of filtering for habitat. It can be "in", "only", "not_in" and "and". See details for more about this argument.
`biome`	(character) The biomes for filtering the dataset. It can be included more than one biome. Default = "All"
`filter_biome`	(character) The type of filtering for biome. It can be "in", "only", "not_in" and "and". See details for more about this argument.
`state`	(character) The states for filtering the dataset. It can be included more than one state. Default = "All".
`filter_state`	(character) The type of filtering for state. It can be "in", "only", "not_in" and "and". See Details for more about this argument.
`vegetation`	(character) The vegetation types for filtering the dataset. It can be included more than one vegetation type. Default = "All".
`filter_vegetation`	(character) The type of filtering for vegetation type. It can be "in", "only", "not_in" and "and". See details for more about this argument.

`endemism`	(character) The endemism (endemic or non-endemic to Brazil) for filtering the dataset. It can be "All", "Endemic" or "Non-endemic". Default = "All".
`origin`	(character) The origin for filtering the dataset. It can be "All", "Native", "Cultivated" and "Naturalized". Default = "All".
`taxonomicStatus`	(character) The taxonomic status for filtering the dataset. It can be "All", "Accepted" or "Synonym". Default = "Accepted".
`nomenclaturalStatus`	(character) The nomenclatural status for filtering the dataset. Default = "Accepted"

Details

It's possible to choose 4 ways to filter by lifeForm, by habitat, by biome, by state and by vegetation type: "in": selects species that have any occurrence of the determined values. It allows multiple matches. For example, if biome = c("Amazon", Cerrado" and filter_biome = "in", it will select all species that occur in the Amazon and Cerrado, some of which may also occur in other biomes.

"only": selects species that have only occurrence of the determined values. It allows only single matches. For example, if biome = c("Amazon", "Cerrado") and filter_biome = "only", it will select all species that occur exclusively in both the Amazon and Cerrado biomes, without any occurrences in other biomes.

"not_in": selects species that don't have occurrence of the determined values. It allows single and multiple matches. For example, if biome = c("Amazon", "Cerrado") and filter_biome = "not_in", it will select all species without occurrences in the Amazon and Cerrado biomes.

"and": selects species that have occurrence in all determined values. It allows single and multiple matches. For example, if biome = c("Amazon", "Cerrado") and filter_biome = "and", it will select all species that occurs only in both the Amazon and Cerrado biomes, including species that occurs in other biomes too.

To get the complete list of arguments available for family, genus, lifeForm, habitat, biome, state, and nomenclaturalStatus, use the function `get_attributes`

Value

A new dataframe with the filtered species.

References

Flora e Funga do Brasil. Jardim Botânico do Rio de Janeiro. Available at: http://floradobrasil.jbrj.gov.br/

Examples

```
data("bf_data") #Load Flora e Funga do Brasil data
#'Select endemic and native species of trees with disjunct occurrence in
# Atlantic Forest and Amazon
am_af_only <- select_species(data = bf_data,
                             include_subspecies = FALSE,
                             include_variety = FALSE,
                             kingdom = "Plantae",
```

```
                                group = "All", subgroup = "All",
                                phylum = "All", class ="All", order = "All",
                                family = "All", genus = "All",
                                lifeForm = "Tree", filter_lifeForm = "only",
                                habitat = "All", filter_habitat = "in",
                                biome = c("Atlantic_Forest","Amazon"),
                                filter_biome = "only",
                                state = "All", filter_state = "and",
                                vegetation = "All",
                                filter_vegetation = "in",
                                endemism = "Endemic", origin = "Native",
                                taxonomicStatus = "All",
                                nomenclaturalStatus = "All")
```

`solve_discrepancies` *Resolve discrepancies between species and subspecies/varieties information*

Description

Resolve discrepancies between species and subspecies/varieties information

Usage

```
solve_discrepancies(data)
```

Arguments

`data`	(data.frame) the data.frame imported with the `load_florabr` function.

Details

In the original dataset, discrepancies may exist between species and subspecies/varieties information. An example of a discrepancy is when a species occurs only in one biome (e.g., Amazon), but a subspecies or variety of the same species occurs in another biome (e.g., Cerrado). This function rectifies such discrepancies by considering distribution (states, biomes, and vegetation), life form, and habitat. For instance, if a subspecies is recorded in a specific biome, it implies that the species also occurs in that biome.

Value

a data.frame with the discrepancies solved

Examples

```
data("bf_data") #Load Flora e Funga do Brasil data
#Check if discrepancies were solved in the dataset
attr(bf_data, "solve_discrepancies")
#Solve discrepancies
```

```
bf_solved <- solve_discrepancies(bf_data)
#Check if discrepancies were solved in the dataset
attr(bf_solved, "solve_discrepancies")
```

`states` *SpatVector of the federal states of Brazil*

Description

A simplified and packed SpatVector of the polygons of the federal states of Brazil. The spatial data was originally obtained from `geobr::read_state`. Borders have been simplified by removing vertices of borders using `terra::simplifyGeom`. It's necessary unpack the Spatvectos using `terra::unwrap`

@usage data(states) states <- terra::unwrap(states)

Usage

```
states
```

Format

A `SpatVector` with 27 geometries and 3 attributes:

abbrev_state State acronym

name_state State's full name

name_region The region to which the state belongs

`subset_species` *Extract a subset of species from Flora e Funga do Brasil database*

Description

Returns a data.frame with a subset of species from Flora e Funga do Brasil database

Usage

```
subset_species(data, species,
                     include_subspecies = FALSE,
                     include_variety = FALSE,
                     kingdom = "Plantae")
```

Arguments

`data`	(data.frame) the data.frame imported with the `load_florabr` function.
`species`	(character) names of the species to be extracted from Flora e Funga do Brasil database.
`include_subspecies`	(logical) include subspecies? Default = FALSE
`include_variety`	(logical) include varieties of the species? Default = FALSE
`kingdom`	(character) The kingdom for filtering the dataset. It can be "Plantae" or "Fungi". Default = "Plantae". To include both, use c("Plantae", "Fungi")

Value

A data.frame with the selected species.

References

Flora e Funga do Brasil. Jardim Botânico do Rio de Janeiro. Available at: http://floradobrasil.jbrj.gov.br/

Examples

```
data("bf_data") #Load Flora e Funga do Brasil data
#Species to extract from database
spp <- c("Araucaria angustifolia", "Adesmia paranensis")
spp_bf <- subset_species(data = bf_data, species = spp,
                      include_subspecies = FALSE,
                      include_variety = FALSE)
spp_bf
```

Index