Package ‘flora’

October 13, 2022

Type Package

Title Tools for Interacting with the Brazilian Flora 2020

Version 0.3.4

Author Gustavo Carvalho

Maintainer Gustavo Carvalho <gustavo.bio@gmail.com>

Depends R (>= 3.0.0)

Imports httr, dplyr, utils, jsonlite, stringdist

Suggests testthat, shiny

Description Tools to quickly compile taxonomic and distribution data from
the Brazilian Flora 2020.

License GPL (>= 2)

URL http://www.github.com/gustavobio/flora

BugReports http://www.github.com/gustavobio/flora/issues

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2020-04-28 05:00:03 UTC

R topics documented:

`df2phytaxa` *Phylomatic format*

Description

Convert the results of get.taxa() to the phylomatic sample format

Usage

```
df2phytaxa(taxa, uppercase = TRUE)
```

Arguments

`taxa`	A data frame with columns named family, genus, and species.
`uppercase`	logical. Should the function capitalize first letters?

`fixCase` *Fix the name case of a taxon*

Description

Fix the name case of a taxon

Usage

```
fixCase(x)
```

Arguments

`x`	a unit character vector with a taxon

Examples

```
fixCase("myrcia lingua")
fixCase("Myrcia Lingua")
fixCase("COFFEA ARABICA")
```

flora *Package flora*

Description

Collect data from the Brazilian Flora checklist (http://floradobrasil.jbrj.gov.br).

Details

This package contains a set of tools solving problems that arise when one has to collect taxonomic and distribution information for large datasets of plants. Interacting with the Brazilian Flora Checklist website from a web browser is often a slow and somewhat cumbersome process, especially if you are not sure about the correct spelling of a name. With flora, however, you can:

- get a suggestion for the correct spelling of a name from an incorrect one
- search for its current taxonomic status
- get its author(s), synonym(s), family, distribution, and lower taxa
- process lists of names and automatically solve synonyms and typing errors

flora now holds all the data it needs. All functions can be used whilst offline.

get.synonyms *List all synonyms of a given taxa*

Description

This function returns all the synonyms of a given taxon according to the Brazilian Flora 2020.

Usage

```
get.synonyms(taxon, fix = FALSE, relationship = FALSE)
```

Arguments

`taxon`	a character vector containing a name.
`fix`	should the function fix issues (synonyms, mispelled names) in taxon?
`relationship`	return the kind of relationship?

Value

a character vector

Examples

```
## Not run:
get.synonyms("Myrcia lingua")

## End(Not run)
```

get.taxa *Get plant taxonomical and distribution data*

Description

This function collects taxonomic information and distribution from the Brazilian Flora Checklist. Synonyms and misspelled names are resolved automatically. Results can be combined with life form, habitat, vernacular name, and occurrence data.

Usage

```
get.taxa(taxa, replace.synonyms = TRUE, suggest.names = TRUE,
  life.form = FALSE, habitat = FALSE, vegetation.type = FALSE,
  vernacular = FALSE, states = FALSE, establishment = FALSE,
  domain = FALSE, endemism = FALSE, drop = c("authorship", "genus",
  "specific.epiteth", "infra.epiteth", "name.status"),
  suggestion.distance = 0.9, parse = FALSE)
```

Arguments

taxa a character vector containing one or more taxa, without authors see remove.authors if you have a list with authorities

replace.synonyms should the function automatically replace synonyms?

suggest.names should the function try to correct misspelled names?

life.form include the life form of the taxon?

habitat include the habitat of the taxon?

vegetation.type include the listed vegetation types?

vernacular include vernacular names and localities?

states include occurrence data?

establishment include the establishment type (native, cultivated or naturalized)?

domain return phytogeographyc domains?

endemism is the taxon endemic to Brazil?

drop NULL or character vector with names of columns with taxonomic information to be removed from the returned data frame. Available names: "id", "scientific.name", "accepted.name", "family", "genus", "specific.epiteth", "infra.epiteth", "taxon.rank", "authorship", "taxon.status", "name.status", "threat.status", and "search.str".

suggestion.distance a value between 0 and 1 indicanting how conservative the name suggestion algorithm should be. Values closer to 1 are very conservative. Be very careful, lower values can give wrong suggestions.

parse Parse names through the GBIF parser to remove authors?

Details

The returned data frame will contain a variable number of rows and columns depending on how the function was called. For instance, since there might be more than one vernacular name for each taxon, some rows will be duplicated if `vernacular` is set to `TRUE`. All misspelled taxa are automatically corrected if the function can come up with a reasonable guess for the name. Conservation status follows the IUCN nomenclature.

Value

a data frame

Examples

```
## Not run:
data(plants)
get.taxa(plants)
get.taxa(plants, life.form = TRUE, establishment = TRUE)

## End(Not run)
```

`get_domains` *Get the phytogeographic domain of a list of taxa*

Description

Get the phytogeographic domain of a list of taxa

Usage

```
get_domains(taxa)
```

Arguments

`taxa`	A data frame with taxon names as returned by get.taxa()

Value

A data frame as returned by get.taxa with an extra column named domain.

Examples

```
## Not run:
taxa <- get.taxa(c("Myrcia guianensis", "bleh", "Xylopia", "Miconia albicans", "bleh", "Poa annua"))
get_domains(taxa)

## End(Not run)
```

get_endemism *Get the listed endemism for a list of taxa*

Description

Get the listed endemism for a list of taxa

Usage

```
get_endemism(taxa)
```

Arguments

`taxa`	A data frame with taxon names as returned by get.taxa()

Value

A data frame as returned by get.taxa with an extra column named vegtype.

Examples

```
## Not run:
taxa <- get.taxa(c("Myrcia guianensis", "bleh", "Xylopia", "Miconia albicans", "bleh", "Poa annua"))
get_endemism(taxa)

## End(Not run)
```

get_lifeform *Get the listed vegetation life forms for a list of taxa*

Description

Get the listed vegetation life forms for a list of taxa

Usage

```
get_lifeform(taxa)
```

Arguments

`taxa`	A data frame with taxon names as returned by get.taxa()

Value

A data frame as returned by get.taxa with an extra column named life.form

Examples

```
## Not run:
taxa <- get.taxa(c("Myrcia guianensis", "bleh", "Xylopia", "Miconia albicans", "bleh", "Poa annua"))
get_lifeform(taxa)

## End(Not run)
```

get_vegtype *Get the listed vegetation types for a list of taxa*

Description

Get the listed vegetation types for a list of taxa

Usage

```
get_vegtype(taxa)
```

Arguments

taxa	A data frame with taxon names as returned by get.taxa()

Value

A data frame as returned by get.taxa with an extra column named vegtype.

Examples

```
## Not run:
taxa <- get.taxa(c("Myrcia guianensis", "bleh", "Xylopia", "Miconia albicans", "bleh", "Poa annua"))
get_vegtype(taxa)

## End(Not run)
```

lower.taxa *Get downstream taxa*

Description

Get all downstream taxa from a family or genus name.

Usage

```
lower.taxa(taxon, accepted = TRUE)
```

Arguments

`taxon`	a character vector with either a family or genus name
`accepted`	list only accepted names?

Examples

```
## Not run:
lower.taxa("Acosmium")
lower.taxa("Zygophyllaceae")

## End(Not run)
```

`occurrence` *Taxa occurrence*

Description

Find the taxa that occur in a given state of Brazil.

Usage

```
occurrence(states, type = c("any", "only", "all"), taxa = NULL)
```

Arguments

`states`	a character vector with one or more state abbreviations following. See notes for abbreviations.
`type`	type of matching to be used. `any` will return the taxa that occur in any of the passed `states`. `only` matches taxa that occur only in all provided (no more, no less) `states` and `all` matches taxa that occur at least in all `states` passed. See examples.
`taxa`	optional character vector to match against the states

Value

a data frame

Note

List of abbreviations: `http://en.wikipedia.org/wiki/States_of_Brazil`

Examples

```
## Not run:
occ.any <- occurrence(c("SP", "BA", "MG"), type = "any")
occ.only <- occurrence(c("SP", "BA", "MG"), type = "only")
occ.all <- occurrence(c("SP", "BA", "MG"), type = "all")
occ.taxa <- occurrence(c("SP", "BA", "MG"), type = "all", taxa = lower.taxa("Myrcia"))

head(occ.any)
head(occ.only)
head(occ.all)
head(occ.taxa)

## End(Not run)
```

`plants` *Plant names*

Description

A small character vector containing 16 plant names. Contains accepted names, synonyms, and misspelled taxa.

Format

A character vector with 16 names

`remove.authors` *Remove the author(s) from a taxon name.*

Description

This attempts to remove the authorities of a taxonomic name.

Usage

```
remove.authors(taxon)
```

Arguments

`taxon`	a character vector containing a single taxon

Value

a character vector

Examples

```
## Not run:
remove.authors("Coffea arabica L.")
remove.authors("Chrysophyllum argenteum subsp. nitidum (G.F.W.Meyer) T.D.Penn.")

## End(Not run)
```

---

standardize.names *Standardize taxonomic names*

Description

This function standardizes taxa names. It is used mainly internally, but might be helpful to the end user in some situations.

Usage

```
standardize.names(taxon)
```

Arguments

taxon	a character vector containing a single name

Value

a character vector

Examples

```
## Not run:
standardize.names("Miconia sp 01")
standardize.names("Miconia Sp 2")
standardize.names("Sp18")

## End(Not run)
```

---

suggest.names *Suggest a valid name from a misspelled one*

Description

This function tries to suggest a valid name according to the Brazilian Flora Checklist using a possibly incorrect one as a starting point.

Usage

```
suggest.names(taxon, max.distance = 0.75, return.na = TRUE,
  ignore.words = NULL)
```

Arguments

`taxon`	a character vector containing a single name
`max.distance`	a numeric value indicating how conservative the function should be when searching for suggestions. Values close to 1 are very conservative
`return.na`	a logical indicating whether to return a `NA` or the original input when no suggestion is found
`ignore.words`	`NULL` or a character vector with words to be ignored by the function. Useful if you are automatizing a workflow and wants the function to ignore words or phrases such as "not found", "dead", "undetermined", and so on

Value

A character vector or `NA`

Examples

```
## Not run:
suggest.names("Cofea arabyca")
suggest.names("Myrcia bela")

## End(Not run)
```

`trim` *Trim a name and remove duplicate tabs and whitespaces*

Description

Remove duplicate and misplaced whitespace characters

Usage

```
trim(taxon)
```

Arguments

`taxon`	a character vector with a single taxon name

Value

a character vector

Examples

```
## Not run:
trim("   Myrcia  lingua")

## End(Not run)
```

vernacular *Vernacular name search*

Description

Search for taxa using vernacular names

Usage

```
vernacular(name, exact = FALSE)
```

Arguments

`name`	a vernacular name
`exact`	approximate or exact match?

Value

a data frame of results or NA

Examples

```
## Not run:
vernacular("pimenta", exact = TRUE)
vernacular("pimenta", exact = FALSE)

## End(Not run)
```

web.flora *Web front end*

Description

This function starts a local webserver to run the shiny app distributed with the package.

Usage

```
web.flora()
```

Index