Package 'BIEN'

January 31, 2025

Title Tools for Accessing the Botanical Information and Ecology Network Database

Version 1.2.7

Description Provides Tools for Accessing the Botanical Information and Ecology Network Database. The BIEN database contains cleaned and standardized botanical data including occurrence, trait, plot and taxonomic data (See <https://bien.nceas.ucsb.edu/bien/> for more Information). This package provides functions that query the BIEN database by constructing and executing optimized SQL queries.

Depends R (>= 4.1.0), RPostgreSQL

License MIT + file LICENSE

Imports DBI, ape, sf, terra, doParallel, parallel, foreach

Suggests knitr, rmarkdown, testthat, maps

VignetteBuilder knitr

RoxygenNote 7.3.2

Encoding UTF-8

NeedsCompilation no

Author Brian Maitner [aut, cre]

Maintainer Brian Maitner <bmaitner@gmail.com>

Repository CRAN

Date/Publication 2025-01-31 22:40:10 UTC

Contents

`BIEN` *BIEN: Tools for accessing the BIEN database.*

Description

The Botanical Information and Ecology Network(BIEN) R package provides access to the BIEN database as well as useful tools for working with the BIEN data.

Getting started

Type vignette("BIEN") to view the vignette, which contains useful information on the BIEN package.

Author(s)

Maintainer: Brian Maitner <bmaitner@gmail.com>

References

Maitner BS, Boyle B, Casler N, et al. The BIEN R package: A tool to access the Botanical Information and Ecology Network (BIEN) Database. Methods Ecol Evol. 2018;9:373-379. https://doi.org/10.1111/2041-210X.12861

`BIEN_list_all` *Extract a list of all species in the BIEN database.*

Description

BIEN_list_all produces a list of all species in the BIEN database.

Usage

```
BIEN_list_all(...)
```

Arguments

...	Additional arguments passed to internal functions.

Value

Dataframe containing a list of all species in the BIEN database.

See Also

Other list functions: BIEN_list_country(), BIEN_list_county(), BIEN_list_sf(), BIEN_list_state()

Examples

```
## Not run:
species_list<-BIEN_list_all()
## End(Not run)
```

BIEN_list_country *Extract species list by country*

Description

BIEN_list_country downloads a list of all species within a country or countries from the BIEN database.

Usage

```
BIEN_list_country(
  country = NULL,
  country.code = NULL,
  cultivated = FALSE,
  new.world = NULL,
  ...
)
```

Arguments

country	A single country or a vector of countries.
country.code	A single country code or a vector of country codes equal in length to the vector of states/province codes.
cultivated	Return information on cultivation status? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
...	Additional arguments passed to internal functions.

Value

Dataframe containing species list(s) for the specified country or countries.

Note

Political division (or political division code) spelling needs to be exact and case-sensitive, see BIEN_metadata_list_political_names for a list of political divisions and associated codes.

See Also

Other list functions: BIEN_list_all(), BIEN_list_county(), BIEN_list_sf(), BIEN_list_state()

Examples

```
## Not run:
BIEN_list_country("Canada")
country_vector<-c("Canada","United States")
BIEN_list_country(country_vector)
## End(Not run)
```

BIEN_list_county *Extract a species list by county.*

Description

BIEN_list_county produces a list of all species with geovalidated occurrences falling within specified county or counties.

Usage

```
BIEN_list_county(
  country = NULL,
  state = NULL,
  county = NULL,
  country.code = NULL,
  state.code = NULL,
  county.code = NULL,
  cultivated = FALSE,
  new.world = NULL,
  ...
)
```

Arguments

country	A single country or vector of countries
state	A state or vector of states (or other primary political divisions, e.g. provinces).
county	A single county (or other secondary administrative boundary)or vector of counties.
country.code	A single country (or other primary administrative boundary) code or a vector of country codes equal in length to the vector of states/province codes.

state.code	A single state/province code, or a vector of states/province codes.
county.code	A single county (or other secondary administrative boundary) code or a vector of county codes equal in length to the vectors of states/province codes and country codes.
cultivated	Return information on cultivation status? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
...	Additional arguments passed to internal functions.

Value

Dataframe containing species list(s) for the specified states/provinces.

Note

Political division (or political division code) spelling needs to be exact and case-sensitive, see BIEN_metadata_list_political_names for a list of political divisions and associated codes.

We recommend using country, state, and county rather than codes, since county names have not been fully standardized.

This function requires you supply either 1) a single state and country with one or more counties, or 2) vectors of equal length for each political level.

See Also

Other list functions: BIEN_list_all(), BIEN_list_country(), BIEN_list_sf(), BIEN_list_state()

Examples

```
## Not run:
BIEN_list_county("United States", "Michigan", "Kent")
BIEN_list_county(country = "United States", state = "Michigan", county = "Kent")
county_vector<-c("Kent","Kalamazoo")
BIEN_list_county(country = "United States", state = "Michigan", county = county_vector)
## End(Not run)
```

BIEN_list_sf *Extract a list of species within a given sf polygon.*

Description

BIEN_list_sf produces a list of all species with occurrence records falling within a user-supplied sf object.

Usage

```
BIEN_list_sf(sf, cultivated = FALSE, new.world = NULL, ...)
```

Arguments

sf	An object of class ff. Note that the object must be in WGS84.
cultivated	Return information on cultivation status? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
...	Additional arguments passed to internal functions.

Value

Dataframe containing a list of all species with occurrences in the supplied sf object.

See Also

Other list functions: BIEN_list_all(), BIEN_list_country(), BIEN_list_county(), BIEN_list_state()

Examples

```
## Not run:
library(sf)

BIEN_ranges_species("Carnegiea gigantea") # saves ranges to the current working directory

sf <- st_read(dsn = ".",
              layer = "Carnegiea_gigantea")

species_list <- BIEN_list_sf(sf = sf)

## End(Not run)
```

BIEN_list_state *Extract a species list by state/province*

Description

BIEN_list_state produces a list of all species with geovalidated occurrences falling within specified state(s) or province(s).

Usage

```
BIEN_list_state(
  country = NULL,
  country.code = NULL,
  state = NULL,
  state.code = NULL,
  cultivated = FALSE,
  new.world = NULL,
  ...
)
```

Arguments

country	A single country or a vector of countries equal in length to the vector of states/provinces.
country.code	A single country code or a vector of country codes equal in length to the vector of states/province codes.
state	A state or vector of states (or other primary political divisions, e.g. provinces).
state.code	A single state/province code, or a vector of states/province codes.
cultivated	Return information on cultivation status? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
...	Additional arguments passed to internal functions.

Value

Dataframe containing species list(s) for the specified states/provinces.

Note

Political division (or political division code) spelling needs to be exact and case-sensitive, see BIEN_metadata_list_political_names for a list of political divisions and associated codes.

See Also

Other list functions: BIEN_list_all(), BIEN_list_country(), BIEN_list_county(), BIEN_list_sf()

Examples

```
## Not run:
BIEN_list_state("United States","Michigan")
state_vector<-c("Michigan","Arizona")
BIEN_list_state(country="United States", state= state_vector)
## End(Not run)
```

BIEN_metadata_citation

Generate citations for data extracted from BIEN.

Description

BIEN_metadata_citation guides a user through the proper documentation for data downloaded from the BIEN database.

Usage

```
BIEN_metadata_citation(
  dataframe = NULL,
  trait.dataframe = NULL,
  trait.mean.dataframe = NULL,
  bibtex_file = NULL,
  acknowledgement_file = NULL,
  ...
)
```

Arguments

dataframe	A data.frame of occurrence data downloaded from the BIEN R package.
trait.dataframe	A data.frame of trait data downloaded from the BIEN R package.
trait.mean.dataframe	A data.frame of species mean trait data from the function BIEN_trait_mean.
bibtex_file	Output file for writing bibtex citations.
acknowledgement_file	Output file for writing acknowledgements.
...	Additional arguments passed to internal functions.

Value

A list object containing information needed for data attribution. Full information for herbaria is available at http://sweetgum.nybg.org/science/ih/

See Also

Other metadata functions: BIEN_metadata_data_dictionaries(), BIEN_metadata_database_version(), BIEN_metadata_list_political_names(), BIEN_metadata_match_data(), BIEN_plot_metadata(), BIEN_ranges_list()

Examples

```
## Not run:
BIEN_metadata_citation()#If you are referencing the phylogeny or range maps.
Xanthium_data<-BIEN_occurrence_species("Xanthium strumarium")
citations<-BIEN_metadata_citation(dataframe=Xanthium_data)#If you are referencing occurrence data
## End(Not run)
```

BIEN_metadata_database_version

Download the current BIEN database version and release date

Description

BIEN_metadata_database_version downloads the current version number and release date for the BIEN database.

Usage

```
BIEN_metadata_database_version(...)
```

Arguments

...	Additional arguments passed to internal functions.

Value

A data frame containing the current version number and release date for the BIEN database.

See Also

Other metadata functions: BIEN_metadata_citation(), BIEN_metadata_data_dictionaries(), BIEN_metadata_list_political_names(), BIEN_metadata_match_data(), BIEN_plot_metadata(), BIEN_ranges_list()

Examples

```
## Not run:
BIEN_metadata_database_version()
## End(Not run)
```

BIEN_metadata_list_political_names

List political divisions and associated geonames codes.

Description

BIEN_metadata_list_political_names downloads country, state, and county names and associated codes used by BIEN.

Usage

```
BIEN_metadata_list_political_names(...)
```

Arguments

...	Additional arguments passed to internal functions.

Value

A dataframe containing political division names and their associated codes.

Note

Political names and codes follow http://www.geonames.org/

See Also

Other metadata functions: BIEN_metadata_citation(), BIEN_metadata_data_dictionaries(), BIEN_metadata_database_version(), BIEN_metadata_match_data(), BIEN_plot_metadata(), BIEN_ranges_list()

Examples

```
## Not run:
BIEN_metadata_list_political_names()
## End(Not run)
```

BIEN_metadata_match_data

Check for differing records between old and new dataframes.

Description

BIEN_metadata_match_data compares old and new dataframes, and can check whether they are identical or be used to select rows that are unique to the old or new versions.

Usage

```
BIEN_metadata_match_data(old, new, return = "identical")
```

Arguments

old	A dataframe that is to be compared to a (typically) newer dataframe.
new	A dataframe that is to be compared to a (typically) older dataframe.
return	What information should be returned? Current options are: "identical" (Logical, are the two dataframes identical?), "additions" (numeric, which rows are new?), "deletions" (numeric, which rows are no longer present?), "logical" (logical, which elements of the old dataframe are in the new one?).

Value

Logical of varying length (depending on choice of "return" parameter)

Note

Since comparisons are done by row (except when using return="identical"), this function may fail to flag additions or deletions if they are exact duplicates of existing rows.

See Also

Other metadata functions: BIEN_metadata_citation(), BIEN_metadata_data_dictionaries(), BIEN_metadata_database_version(), BIEN_metadata_list_political_names(), BIEN_plot_metadata(), BIEN_ranges_list()

Examples

```
## Not run:
new<-BIEN_occurrence_species("Acer nigrum")
old<-new[-1:-4,]#simulate having an older dataset by removing four rows
BIEN_metadata_match_data(old,new,return="identical")
BIEN_metadata_match_data(old,new,return="additions")
## End(Not run)
```

BIEN_occurrence_box *Extract species occurrence records by a latitude/longitude bounding box.*

Description

BIEN_occurrence_box extracts occurrences records falling within the specific area.

Usage

```
BIEN_occurrence_box(
  min.lat,
  max.lat,
  min.long,
  max.long,
  species = NULL,
  genus = NULL,
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  observation.type = FALSE,
  political.boundaries = TRUE,
  collection.info = FALSE,
  ...
)
```

Arguments

min.lat	Minimum latitude
max.lat	Maximum latitude
min.long	Minimum longitude
max.long	Maximum longitude
species	Optional. A single species or a vector of species.
genus	Optional. A single genus or a vector of genera.
cultivated	Return known cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.
natives.only	Exclude detected introduced species? Default is TRUE.
observation.type	Return information on type of observation (i.e. specimen vs. plot)? The default value is FALSE.
political.boundaries	Return information on political boundaries for an observation? The default value is FALSE.
collection.info	Return additional information about collection and identification? The default value is FALSE.
...	Additional arguments passed to internal functions.

Value

Dataframe containing occurrence records for the specified area.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

Specifying species and/or genera will limit records returned to that set of taxa.

See Also

Other occurrence functions: BIEN_occurrence_country(), BIEN_occurrence_county(), BIEN_occurrence_family(), BIEN_occurrence_genus(), BIEN_occurrence_records_per_species(), BIEN_occurrence_sf(), BIEN_occurrence_species(), BIEN_occurrence_state()

Examples

```
## Not run:
output_test<-
BIEN_occurrence_box(min.lat = 32,max.lat = 33,min.long = -114,max.long = -113,
cultivated = TRUE, new.world = FALSE)
## End(Not run)
```

BIEN_occurrence_country

Extract species occurrence records by country.

Description

BIEN_occurrence_country extracts occurrences records for the specified country/countries.

Usage

```
BIEN_occurrence_country(
  country = NULL,
  country.code = NULL,
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  observation.type = FALSE,
  political.boundaries = FALSE,
  collection.info = FALSE,
  only.geovalid = TRUE,
  ...
)
```

Arguments

country	A single country or a vector of country.
country.code	A single country code or a vector of country codes equal in length to the vector of states/province codes.
cultivated	Return known cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.
natives.only	Exclude detected introduced species? Default is TRUE.

observation.type
: Return information on type of observation (i.e. specimen vs. plot)? The default value is FALSE.

political.boundaries
: Return information on political boundaries for an observation? The default value is FALSE.

collection.info
: Return additional information about collection and identification? The default value is FALSE.

only.geovalid
: Should the returned records be limited to those with validated coordinates? Default is TRUE

...
: Additional arguments passed to internal functions.

Value

Dataframe containing occurrence records for the specified country.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

Political division (or political division code) spelling needs to be exact and case-sensitive, see BIEN_metadata_list_political_names for a list of political divisions and associated codes.

See Also

Other occurrence functions: BIEN_occurrence_box(), BIEN_occurrence_county(), BIEN_occurrence_family(), BIEN_occurrence_genus(), BIEN_occurrence_records_per_species(), BIEN_occurrence_sf(), BIEN_occurrence_species(), BIEN_occurrence_state()

Examples

```
## Not run:
BIEN_occurrence_country("Cuba")
country_vector<-c("Cuba","Bahamas")
BIEN_occurrence_country(country_vector)
## End(Not run)
```

BIEN_occurrence_county

Extract species occurrence records by county.

Description

BIEN_occurrence_county extracts occurrences records for the specified county or counties.

Usage

```
BIEN_occurrence_county(
  country = NULL,
  state = NULL,
  county = NULL,
  country.code = NULL,
  state.code = NULL,
  county.code = NULL,
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  observation.type = FALSE,
  political.boundaries = FALSE,
  collection.info = FALSE,
  ...
)
```

Arguments

country	A single country or vector of countries.
state	A state or vector of states (or other primary political divisions, e.g. provinces).
county	A single county or a vector of counties (or other secondary political division, e.g. parish).
country.code	A single country (or other primary administrative boundary) code or a vector of country codes equal in length to the vector of states/province codes.
state.code	A single state/province code, or a vector of states/province codes.
county.code	A single county (or other secondary administrative boundary) code or a vector of county codes equal in length to the vectors of states/province codes and country codes.
cultivated	Return known cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.
natives.only	Exclude detected introduced species? Default is TRUE.
observation.type	Return information on type of observation (i.e. specimen vs. plot)? The default value is FALSE.
political.boundaries	Return information on political boundaries for an observation? The default value is FALSE.

collection.info
: Return additional information about collection and identification? The default value is FALSE.

...
: Additional arguments passed to internal functions.

Value

Dataframe containing occurrence records for the specified states/provinces.

Note

Political division (or political division code) spelling needs to be exact and case-sensitive, see BIEN_metadata_list_political_names for a list of political divisions and associated codes.

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

This function requires you supply either 1) a single country with one or more states, or 2) vectors of equal length for each political level.

See Also

Other occurrence functions: BIEN_occurrence_box(), BIEN_occurrence_country(), BIEN_occurrence_family(), BIEN_occurrence_genus(), BIEN_occurrence_records_per_species(), BIEN_occurrence_sf(), BIEN_occurrence_species(), BIEN_occurrence_state()

Examples

```
## Not run:
BIEN_occurrence_county("United States","Arizona","Pima")
country_vector<-c("United States","United States")
state_vector<-c("Arizona","Michigan")
county_vector<-c("Pima","Kent")
BIEN_occurrence_county(country=country_vector, state = state_vector, county = county_vector)
## End(Not run)
```

BIEN_occurrence_family

Extract species occurrences by family.

Description

BIEN_occurrence_family extracts all occurrences for a given family (or families) from the BIEN database.

Usage

```
BIEN_occurrence_family(
  family,
  cultivated = FALSE,
  new.world = NULL,
  observation.type = FALSE,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  political.boundaries = FALSE,
  collection.info = FALSE,
  only.geovalid = TRUE,
  ...
)
```

Arguments

family	A single family or a vector of families.
cultivated	Return known cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
observation.type	Return information on type of observation (i.e. specimen vs. plot)? The default value is FALSE.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.
natives.only	Exclude detected introduced species? Default is TRUE.
political.boundaries	Return information on political boundaries for an observation? The default value is FALSE.
collection.info	Return additional information about collection and identification? The default value is FALSE.
only.geovalid	Should the returned records be limited to those with validated coordinates? Default is TRUE
...	Additional arguments passed to internal functions.

Value

Dataframe containing occurrence records for the specified family/families.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

See Also

Other occurrence functions: BIEN_occurrence_box(), BIEN_occurrence_country(), BIEN_occurrence_county(), BIEN_occurrence_genus(), BIEN_occurrence_records_per_species(), BIEN_occurrence_sf(), BIEN_occurrence_species(), BIEN_occurrence_state()

Examples

```
## Not run:
BIEN_occurrence_family("Theaceae")
family_vector<-c("Theaceae","Ericaceae")
BIEN_occurrence_family(family_vector)
## End(Not run)
```

BIEN_occurrence_genus *Extract occurrence data from BIEN for specified genera*

Description

BIEN_occurrence_genus downloads occurrence records for specific genus/genera from the BIEN database.

Usage

```
BIEN_occurrence_genus(
  genus,
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  observation.type = FALSE,
  political.boundaries = FALSE,
  collection.info = FALSE,
  ...
)
```

Arguments

genus	A single genus, or a vector of genera. Genera should be capitalized.
cultivated	Return known cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.

natives.only Exclude detected introduced species? Default is TRUE.

observation.type
: Return information on type of observation (i.e. specimen vs. plot)? The default value is FALSE.

political.boundaries
: Return information on political boundaries for an observation? The default value is FALSE.

collection.info
: Return additional information about collection and identification? The default value is FALSE.

... Additional arguments passed to internal functions.

Value

Dataframe containing occurrence records for the specified genera.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

See Also

Other occurrence functions: BIEN_occurrence_box(), BIEN_occurrence_country(), BIEN_occurrence_county(), BIEN_occurrence_family(), BIEN_occurrence_records_per_species(), BIEN_occurrence_sf(), BIEN_occurrence_species(), BIEN_occurrence_state()

Examples

```
## Not run:
BIEN_occurrence_genus("Abutilon")
genus_vector<-c("Abutilon","Abronia")
BIEN_occurrence_genus(genus_vector)
BIEN_occurrence_genus(genus = "Abutilon",cultivated = TRUE,new.world = FALSE)
## End(Not run)
```

BIEN_occurrence_records_per_species

Count the number of (geoValid) occurrence records for each species in BIEN

Description

BIEN_occurrence_records_per_species downloads a count of the number of geovalidated occurrence records for each species in the BIEN database.

Usage

```
BIEN_occurrence_records_per_species(species = NULL, ...)
```

Arguments

species	A single species, or vector of species. If NULL, the default, it will return counts for all species.
...	Additional arguments passed to internal functions.

Value

A dataframe listing the number of geovalidated occurrence records for each species in the BIEN database.

See Also

Other occurrence functions: BIEN_occurrence_box(), BIEN_occurrence_country(), BIEN_occurrence_county(), BIEN_occurrence_family(), BIEN_occurrence_genus(), BIEN_occurrence_sf(), BIEN_occurrence_species(), BIEN_occurrence_state()

Examples

```
## Not run:
occurrence_counts<-BIEN_occurrence_records_per_species()
## End(Not run)
```

BIEN_occurrence_sf *Extract occurrence data for specified sf polygon*

Description

BIEN_occurrence_sf downloads occurrence records falling within a user-specified sf polygon

Usage

```
BIEN_occurrence_sf(
  sf,
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  observation.type = FALSE,
  political.boundaries = FALSE,
  collection.info = FALSE,
  ...
)
```

Arguments

sf	An object of class sf. Note that the projection must be WGS84.
cultivated	Return known cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.
natives.only	Exclude detected introduced species? Default is TRUE.
observation.type	Return information on type of observation (i.e. specimen vs. plot)? The default value is FALSE.
political.boundaries	Return information on political boundaries for an observation? The default value is FALSE.
collection.info	Return additional information about collection and identification? The default value is FALSE.
...	Additional arguments passed to internal functions.

Value

Dataframe containing occurrence records falling within the polygon.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

See Also

Other occurrence functions: BIEN_occurrence_box(), BIEN_occurrence_country(), BIEN_occurrence_county(), BIEN_occurrence_family(), BIEN_occurrence_genus(), BIEN_occurrence_records_per_species(), BIEN_occurrence_species(), BIEN_occurrence_state()

Examples

```
## Not run:
library(sf)

# first, we download an example shapefile to use (a species range)

BIEN_ranges_species("Carnegiea gigantea")#saves range to the current working directory

# load the range map as an sf object
```

```
sf <- st_read(dsn = ".",layer = "Carnegiea_gigantea")

# get the occurrences that occur within the polygon.

species_occurrences <- BIEN_occurrence_sf(sf = sf)

## End(Not run)
```

BIEN_occurrence_species *Extract occurrence data for specified species from BIEN*

Description

BIEN_occurrence_species downloads occurrence records for specific species from the BIEN database.

Usage

```
BIEN_occurrence_species(
  species,
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  observation.type = FALSE,
  political.boundaries = FALSE,
  collection.info = FALSE,
  only.geovalid = TRUE,
  ...
)
```

Arguments

species	A single species, or a vector of species. Genus and species should be separated by a space. Genus should be capitalized.
cultivated	Return known cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.
natives.only	Exclude detected introduced species? Default is TRUE.
observation.type	Return information on type of observation (i.e. specimen vs. plot)? The default value is FALSE.

`political.boundaries`
: Return information on political boundaries for an observation? The default value is FALSE.

`collection.info`
: Return additional information about collection and identification? The default value is FALSE.

`only.geovalid`
: Should the returned records be limited to those with validated coordinates? Default is TRUE

`...`
: Additional arguments passed to internal functions.

Value

Dataframe containing occurrence records for the specified species.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

See Also

Other occurrence functions: `BIEN_occurrence_box()`, `BIEN_occurrence_country()`, `BIEN_occurrence_county()`, `BIEN_occurrence_family()`, `BIEN_occurrence_genus()`, `BIEN_occurrence_records_per_species()`, `BIEN_occurrence_sf()`, `BIEN_occurrence_state()`

Examples

```
## Not run:
BIEN_occurrence_species("Abies amabilis")
species_vector<-c("Abies amabilis", "Acer nigrum")
BIEN_occurrence_species(species_vector)
BIEN_occurrence_species(species_vector,all.taxonomy = TRUE)
## End(Not run)
```

BIEN_occurrence_state *Extract species occurrence records by state.*

Description

BIEN_occurrence_state extracts occurrences records for the specified state(s).

Usage

```
BIEN_occurrence_state(
  country = NULL,
  state = NULL,
  country.code = NULL,
  state.code = NULL,
```

```
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  observation.type = FALSE,
  political.boundaries = FALSE,
  collection.info = FALSE,
  ...
)
```

Arguments

country	A single country or vector of countries.
state	A state or vector of states (or other primary political divisions, e.g. provinces).
country.code	A single country (or other primary administrative boundary) code or a vector of country codes equal in length to the vector of states/province codes.
state.code	A single state/province code, or a vector of states/province codes.
cultivated	Return known cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.
natives.only	Exclude detected introduced species? Default is TRUE.
observation.type	Return information on type of observation (i.e. specimen vs. plot)? The default value is FALSE.
political.boundaries	Return information on political boundaries for an observation? The default value is FALSE.
collection.info	Return additional information about collection and identification? The default value is FALSE.
...	Additional arguments passed to internal functions.

Value

Dataframe containing occurrence records for the specified states/provinces.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

Political division (or political division code) spelling needs to be exact and case-sensitive, see BIEN_metadata_list_political_names for a list of political divisions and associated codes.

This function requires you supply either 1) a single country with one or more states, or 2) vectors of equal length for each political level.

See Also

Other occurrence functions: BIEN_occurrence_box(), BIEN_occurrence_country(), BIEN_occurrence_county(), BIEN_occurrence_family(), BIEN_occurrence_genus(), BIEN_occurrence_records_per_species(), BIEN_occurrence_sf(), BIEN_occurrence_species()

Examples

```
## Not run:
BIEN_occurrence_state("United States","Rhode Island")
state_vector<-c("Rhode Island","Maryland")
BIEN_occurrence_state(country="United States",state=state_vector)
## End(Not run)
```

BIEN_phylogeny_complete

Download the complete BIEN phylogenies

Description

BIEN_phylogeny_complete downloads a specified number of the BIEN phylogeny replicates.

Usage

```
BIEN_phylogeny_complete(n_phylogenies = 1, seed = NULL, replicates = NULL, ...)
```

Arguments

n_phylogenies	The number of phylogenies to download. Should be an integer between 1 and 100. Default is 1.
seed	Argument passed to set.seed. Useful for replicating work with random phylogeny sets.
replicates	The specific replicated phylogenies to return. Should be a numeric vector of integers between 1 and 100.
...	Additional arguments passed to internal functions.

Value

A phylo or multiphylo object containing the specified phylogenies

Note

Information on the construction of the BIEN phylogenies is available online at https://bien.nceas.ucsb.edu/bien/biendata/bien-2/phylogeny/

See Also

Other phylogeny functions: BIEN_phylogeny_conservative(), BIEN_phylogeny_label_nodes()

Examples

```
## Not run:
phylos<-BIEN_phylogeny_complete(n_phylogenies = 10,seed = 1)
phylos<-BIEN_phylogeny_complete(replicates = c(1,2,99,100))
## End(Not run)
```

BIEN_phylogeny_conservative

Download the conservative BIEN phylogeny

Description

BIEN_phylogeny_conservative downloads the conservative BIEN phylogeny, which only includes species with molecular data available.

Usage

```
BIEN_phylogeny_conservative(...)
```

Arguments

...	Additional arguments passed to internal functions.

Value

A phylo object containing the BIEN conservative phylogeny

Note

Information on the construction of the BIEN phylogenies is available online at https://bien.nceas.ucsb.edu/bien/biendata/bien-2/phylogeny/

See Also

Other phylogeny functions: BIEN_phylogeny_complete(), BIEN_phylogeny_label_nodes()

Examples

```
## Not run:
BIEN_phylo<-BIEN_phylogeny_conservative()
## End(Not run)
```

BIEN_phylogeny_label_nodes *Label nodes on a phylogeny*

Description

BIEN_phylogeny_label_nodes will label the nodes on a phylogeny based on either the BIEN taxonomy or user-supplied taxa.

Usage

```
BIEN_phylogeny_label_nodes(
  phylogeny,
  family = TRUE,
  genus = FALSE,
  other_taxa = NULL,
  ...
)
```

Arguments

phylogeny	A single phylogeny.
family	Should family-level nodes be labeled? Default is TRUE.
genus	Should genus-level nodes be labeled? Default is FALSE. Overwrites family-level nodes where a family contains a single genera.
other_taxa	A dataframe containing two columns: 1) the taxa to be labelled; 2) the species associated with each taxon.
...	Additional arguments passed to internal functions.

Value

Input phylogeny with labeled nodes.

Note

Information on the construction of the BIEN phylogenies is available online at https://bien.nceas.ucsb.edu/bien/biendata/bien-2/phylogeny/

See Also

Other phylogeny functions: BIEN_phylogeny_complete(), BIEN_phylogeny_conservative()

Examples

```
## Not run:
phylogeny<-BIEN_phylogeny_conservative()

phylogeny<-drop.tip(phy = phylogeny,tip = 101:length(phylogeny$tip.label))
plot.phylo(x = phylogeny,show.tip.label = FALSE)

fam_nodes<-BIEN_phylogeny_label_nodes(phylogeny = phylogeny,family = TRUE)
plot.phylo(x = fam_nodes,show.tip.label = FALSE, show.node.label = TRUE)

gen_nodes<-BIEN_phylogeny_label_nodes(phylogeny = phylogeny, family = FALSE, genus = TRUE)
plot.phylo(x = gen_nodes, show.tip.label = FALSE, show.node.label = TRUE)

other_taxa <- as.data.frame(matrix(nrow = 10,ncol = 2))
colnames(other_taxa)<-c("taxon","species")
other_taxa$taxon[1:5]<-"A" #Randomly assign a few species to taxon A
other_taxa$taxon[6:10]<-"B" #Randomly assign a few species to taxon B
tax_nodes <-
 BIEN_phylogeny_label_nodes(phylogeny = phylogeny,
                            family = FALSE, genus = FALSE, other_taxa = other_taxa)
plot.phylo(x = tax_nodes,show.tip.label = FALSE,show.node.label = TRUE)
## End(Not run)
```

BIEN_plot_country *Download plot data from specified countries.*

Description

BIEN_plot_country downloads all plot data from specified countries.

Usage

```
BIEN_plot_country(
  country = NULL,
  country.code = NULL,
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  political.boundaries = FALSE,
  collection.info = FALSE,
  all.metadata = FALSE,
  ...
)
```

Arguments

country	A country or vector of countries.
country.code	A single country code or a vector of country codes equal in length to the vector of states/province codes.
cultivated	Return cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.
natives.only	Exclude detected introduced species? Default is TRUE.
political.boundaries	Return information on political boundaries for an observation? The default value is FALSE.
collection.info	Return additional information about collection and identification? The default value is FALSE.
all.metadata	Should additional plot metadata be returned? Default is FALSE.
...	Additional arguments passed to internal functions.

Value

A dataframe containing all data from the specified countries.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

Political division (or political division code) spelling needs to be exact and case-sensitive, see BIEN_metadata_list_political_names for a list of political divisions and associated codes.

See Also

Other plot functions: BIEN_plot_dataset(), BIEN_plot_datasource(), BIEN_plot_list_datasource(), BIEN_plot_list_sampling_protocols(), BIEN_plot_metadata(), BIEN_plot_name(), BIEN_plot_sampling_protoco
BIEN_plot_sf(), BIEN_plot_state()

Examples

```
## Not run:
BIEN_plot_country("Costa Rica")
BIEN_plot_country(c("Costa Rica","Panama"))
## End(Not run)
```

BIEN_plot_dataset *Download plot data by dataset.*

Description

BIEN_plot_dataset downloads all plot data for a given dataset or datasets.

Usage

```
BIEN_plot_dataset(
  dataset,
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  political.boundaries = FALSE,
  collection.info = FALSE,
  all.metadata = FALSE,
  ...
)
```

Arguments

dataset	A plot dataset or vector of datasets. See BIEN_plot_metadata for more information on plots.
cultivated	Return cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.
natives.only	Exclude detected introduced species? Default is TRUE.
political.boundaries	Return information on political boundaries for an observation? The default value is FALSE.
collection.info	Return additional information about collection and identification? The default value is FALSE.
all.metadata	Should additional plot metadata be returned? Default is FALSE.
...	Additional arguments passed to internal functions.

Value

A dataframe containing all data from the specified dataset.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

Datasets and related information can be looked up with BIEN_plot_metadata

See Also

Other plot functions: BIEN_plot_country(), BIEN_plot_datasource(), BIEN_plot_list_datasource(), BIEN_plot_list_sampling_protocols(), BIEN_plot_metadata(), BIEN_plot_name(), BIEN_plot_sampling_protoco BIEN_plot_sf(), BIEN_plot_state()

Examples

```
## Not run:
BIEN_plot_dataset("Gentry Transect Dataset")
## End(Not run)
```

BIEN_plot_datasource *Download plot data from a given datasource.*

Description

BIEN_plot_datasource downloads all plot data from a given datasource.

Usage

```
BIEN_plot_datasource(
  datasource,
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  political.boundaries = FALSE,
  collection.info = FALSE,
  all.metadata = FALSE,
  ...
)
```

Arguments

datasource	A datasource. See BIEN_plot_list_datasource for options.
cultivated	Return cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.

all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.
natives.only	Exclude detected introduced species? Default is TRUE.
political.boundaries	Return information on political boundaries for an observation? The default value is FALSE.
collection.info	Return additional information about collection and identification? The default value is FALSE.
all.metadata	Should additional plot metadata be returned? Default is FALSE.
...	Additional arguments passed to internal functions.

Value

A dataframe containing all data from the specified datasource.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

See Also

Other plot functions: BIEN_plot_country(), BIEN_plot_dataset(), BIEN_plot_list_datasource(), BIEN_plot_list_sampling_protocols(), BIEN_plot_metadata(), BIEN_plot_name(), BIEN_plot_sampling_protoco
BIEN_plot_sf(), BIEN_plot_state()

Examples

```
## Not run:
BIEN_plot_datasource("SALVIAS")
## End(Not run)
```

BIEN_plot_list_datasource

List available datasources.

Description

BIEN_plot_list_datasource list all plot datasources in the BIEN database.

Usage

```
BIEN_plot_list_datasource(...)
```

Arguments

...	Additional arguments passed to internal functions.

Value

A vector of available datasources.

See Also

Other plot functions: BIEN_plot_country(), BIEN_plot_dataset(), BIEN_plot_datasource(), BIEN_plot_list_sampling_protocols(), BIEN_plot_metadata(), BIEN_plot_name(), BIEN_plot_sampling_protoco BIEN_plot_sf(), BIEN_plot_state()

Examples

```
## Not run:
BIEN_plot_list_datasource()
## End(Not run)
```

BIEN_plot_list_sampling_protocols

List available sampling protocols.

Description

BIEN_plot_list_sampling_protocols list all available sampling protocols.

Usage

```
BIEN_plot_list_sampling_protocols(...)
```

Arguments

...	Additional arguments passed to internal functions.

Value

A vector of available sampling protocols.

See Also

Other plot functions: BIEN_plot_country(), BIEN_plot_dataset(), BIEN_plot_datasource(), BIEN_plot_list_datasource(), BIEN_plot_metadata(), BIEN_plot_name(), BIEN_plot_sampling_protocol(), BIEN_plot_sf(), BIEN_plot_state()

Examples

```
## Not run:
BIEN_plot_list_sampling_protocols()
## End(Not run)
```

BIEN_plot_metadata *Download plot metadata*

Description

BIEN_plot_metadata downloads the plot metadata table.

Usage

```
BIEN_plot_metadata(...)
```

Arguments

...	Additional arguments passed to internal functions.

Value

A dataframe containing plot metadata.

See Also

Other plot functions: BIEN_plot_country(), BIEN_plot_dataset(), BIEN_plot_datasource(), BIEN_plot_list_datasource(), BIEN_plot_list_sampling_protocols(), BIEN_plot_name(), BIEN_plot_sampling_protocol(), BIEN_plot_sf(), BIEN_plot_state()

Other metadata functions: BIEN_metadata_citation(), BIEN_metadata_data_dictionaries(), BIEN_metadata_database_version(), BIEN_metadata_list_political_names(), BIEN_metadata_match_data(), BIEN_ranges_list()

Examples

```
## Not run:
BIEN_plot_metadata()
## End(Not run)
```

BIEN_plot_name *Download plot data by plot name.*

Description

BIEN_plot_name downloads all plot data for a set of plot names.

Usage

```
BIEN_plot_name(
  plot.name,
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  political.boundaries = FALSE,
  collection.info = FALSE,
  all.metadata = FALSE,
  ...
)
```

Arguments

plot.name	A plot name or vector of names. See BIEN_plot_metadata for more information on plots.
cultivated	Return cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.
natives.only	Exclude detected introduced species? Default is TRUE.
political.boundaries	Return information on political boundaries for an observation? The default value is FALSE.
collection.info	Return additional information about collection and identification? The default value is FALSE.
all.metadata	Should additional plot metadata be returned? Default is FALSE.
...	Additional arguments passed to internal functions.

Value

A dataframe containing all data from the specified plot(s).

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

Plot names can be looked up with BIEN_plot_metadata.

See Also

Other plot functions: BIEN_plot_country(), BIEN_plot_dataset(), BIEN_plot_datasource(), BIEN_plot_list_datasource(), BIEN_plot_list_sampling_protocols(), BIEN_plot_metadata(), BIEN_plot_sampling_protocol(), BIEN_plot_sf(), BIEN_plot_state()

Examples

```
## Not run:
BIEN_plot_name("SR-1")
## End(Not run)
```

BIEN_plot_sampling_protocol

Download plot data using a specified sampling protocol.

Description

BIEN_plot_sampling_protocol downloads all plot data using a specified sampling protocol.

Usage

```
BIEN_plot_sampling_protocol(
  sampling_protocol,
  cultivated = FALSE,
  new.world = FALSE,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  political.boundaries = FALSE,
  collection.info = FALSE,
  all.metadata = FALSE,
  ...
)
```

Arguments

sampling_protocol	A sampling protocol or vector of sampling protocols. See BIEN_plot_list_sampling_protocols for options.
cultivated	Return cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.

natives.only	Exclude detected introduced species? Default is TRUE.
political.boundaries	Return information on political boundaries for an observation? The default value is FALSE.
collection.info	Return additional information about collection and identification? The default value is FALSE.
all.metadata	Should additional plot metadata be returned? Default is FALSE.
...	Additional arguments passed to internal functions.

Value

A dataframe containing all data from the specified datasource.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

See Also

Other plot functions: BIEN_plot_country(), BIEN_plot_dataset(), BIEN_plot_datasource(), BIEN_plot_list_datasource(), BIEN_plot_list_sampling_protocols(), BIEN_plot_metadata(), BIEN_plot_name(), BIEN_plot_sf(), BIEN_plot_state()

Examples

```
## Not run:
BIEN_plot_sampling_protocol("Point-intercept")
## End(Not run)
```

BIEN_plot_sf *Download plot data from specified sf object.*

Description

BIEN_plot_sf downloads all plot data falling within a supplied sf polygon.

Usage

```
BIEN_plot_sf(
  sf,
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
```

```
  political.boundaries = TRUE,
  collection.info = FALSE,
  all.metadata = FALSE,
  ...
)
```

Arguments

sf	An object of class sf. Note that the projection must be WGS84.
cultivated	Return cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.
natives.only	Exclude detected introduced species? Default is TRUE.
political.boundaries	Return information on political boundaries for an observation? The default value is FALSE.
collection.info	Return additional information about collection and identification? The default value is FALSE.
all.metadata	Should additional plot metadata be returned? Default is FALSE.
...	Additional arguments passed to internal functions.

Value

A dataframe containing all plot data from within the specified sf polygon.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

See Also

Other plot functions: BIEN_plot_country(), BIEN_plot_dataset(), BIEN_plot_datasource(), BIEN_plot_list_datasource(), BIEN_plot_list_sampling_protocols(), BIEN_plot_metadata(), BIEN_plot_name(), BIEN_plot_sampling_protocol(), BIEN_plot_state()

Examples

```
## Not run:
library(sf)

BIEN_ranges_species("Carnegiea gigantea") #saves ranges to the current working directory
```

```
sf <- st_read(dsn = ".",
              layer = "Carnegiea_gigantea")

saguaro_plot_data <- BIEN_plot_sf(sf = sf)

## End(Not run)
```

BIEN_plot_state *Download plot data from specified states/provinces.*

Description

BIEN_plot_state downloads all plot data from specified states/provinces.

Usage

```
BIEN_plot_state(
  country = NULL,
  state = NULL,
  country.code = NULL,
  state.code = NULL,
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  political.boundaries = TRUE,
  collection.info = FALSE,
  all.metadata = FALSE,
  ...
)
```

Arguments

country	A single country.
state	A state or vector of states (or other primary political divisions).
country.code	A single country code or a vector of country codes equal in length to the vector of states/province codes.
state.code	A single state/province code, or a vector of states/province codes.
cultivated	Return cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.

natives.only	Exclude detected introduced species? Default is TRUE.
political.boundaries	Return information on political boundaries for an observation? The default value is FALSE.
collection.info	Return additional information about collection and identification? The default value is FALSE.
all.metadata	Should additional plot metadata be returned? Default is FALSE.
...	Additional arguments passed to internal functions.

Value

A dataframe containing all data from the specified states.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

Political division (or political division code) spelling needs to be exact and case-sensitive, see BIEN_metadata_list_political_names for a list of political divisions and associated codes.

This function requires you supply either 1) a single country with one or states, or 2) vectors of equal length for each political level.

See Also

Other plot functions: BIEN_plot_country(), BIEN_plot_dataset(), BIEN_plot_datasource(), BIEN_plot_list_datasource(), BIEN_plot_list_sampling_protocols(), BIEN_plot_metadata(), BIEN_plot_name(), BIEN_plot_sampling_protocol(), BIEN_plot_sf()

Examples

```
## Not run:
BIEN_plot_state(country="United States", state="Colorado")
BIEN_plot_state(country="United States",state= c("Colorado","California"))
## End(Not run)
```

BIEN_ranges_box *Download range maps that intersect a specified bounding box.*

Description

BIEN_ranges_box extracts range maps for a specified bounding box.

Usage

```
BIEN_ranges_box(
  min.lat,
  max.lat,
  min.long,
  max.long,
  directory = NULL,
  species.names.only = FALSE,
  return.species.list = TRUE,
  crop.ranges = FALSE,
  include.gid = FALSE,
  ...
)
```

Arguments

min.lat	Minimum latitude of the ranges included.
max.lat	Maximum latitude of the ranges included.
min.long	Minimum longitude of the ranges included.
max.long	Maximum longitude of the ranges included.
directory	Directory that range maps should be saved in. If none is specified, range maps will be saved in the current working directory.
species.names.only	Return species names rather than spatial data? Default is FALSE.
return.species.list	Should a species list be returned in addition to downloading range maps? Default is FALSE
crop.ranges	Should the ranges be cropped to the focal area? Default is FALSE.
include.gid	Should the files returned have a unique GID appended to them? This is needed if downloading multiple maps for the same species.
...	Additional arguments passed to internal functions.

Value

Range maps for all available species within the specified bounding box.

Note

Details on the construction of BIEN range maps is available at https://bien.nceas.ucsb.edu/bien/biendata/bien-3/

See Also

Other range functions: BIEN_ranges_genus(), BIEN_ranges_intersect_species(), BIEN_ranges_list(), BIEN_ranges_load_species(), BIEN_ranges_sf(), BIEN_ranges_shapefile_to_skinny(), BIEN_ranges_skinny_ra
BIEN_ranges_species(), BIEN_ranges_species_bulk()

Examples

```
## Not run:
temp_dir <- file.path(tempdir(), "BIEN_temp") #Set a working directory
BIEN_ranges_box(42,43,-85,-84,species.names.only = TRUE)
BIEN_ranges_box(42,43,-85,-84,directory = temp_dir)
## End(Not run)
```

BIEN_ranges_genus *Download range maps for given genus.*

Description

BIEN_ranges_genus extracts range maps for the specified genera.

Usage

```
BIEN_ranges_genus(
  genus,
  directory = NULL,
  matched = TRUE,
  match_names_only = FALSE,
  include.gid = FALSE,
  ...
)
```

Arguments

genus	A single genus or a vector of genera.
directory	Directory that range maps should be saved in. If none is specified, range maps will be saved in the current working directory.
matched	Return a list of taxa that were downloaded. Default is TRUE.
match_names_only	Check for range maps for the taxa specified without downloading range maps. Default is FALSE.
include.gid	Should the files returned have a unique GID appended to them? This is needed if downloading multiple maps for the same species.
...	Additional arguments passed to internal functions.

Value

Range maps for all available species within the specified genera.

Note

Details on the construction of BIEN range maps is available at https://bien.nceas.ucsb.edu/bien/biendata/bien-3/

See Also

Other range functions: BIEN_ranges_box(), BIEN_ranges_intersect_species(), BIEN_ranges_list(), BIEN_ranges_load_species(), BIEN_ranges_sf(), BIEN_ranges_shapefile_to_skinny(), BIEN_ranges_skinny_ra
BIEN_ranges_species(), BIEN_ranges_species_bulk()

Examples

```
## Not run:
library(maps)
library(sf)

genus_vector <- c("Abies","Acer")

temp_dir <- file.path(tempdir(), "BIEN_temp")#Set a working directory

BIEN_ranges_genus(genus_vector)

BIEN_ranges_genus(genus = genus_vector,
                  match_names_only = TRUE)

BIEN_ranges_genus(genus = genus_vector,
                  directory = temp_dir) #saves ranges to a specified working directory

BIEN_ranges_genus("Abies")

BIEN_ranges_genus(genus = "Abies",
                  directory = temp_dir)

#Reading files

Abies_poly <- read_sf(dsn = temp_dir,layer = "Abies_lasiocarpa")

#Plotting files

plot(Abies_poly[1]) #plots the range, but doesn't mean much without any reference

map('world', fill = TRUE, col = "grey") #plots a world map (WGS84 projection), in grey

plot(Abies_poly[1],
     col="forest green",
     add = TRUE) #adds the range of Abies lasiocarpa to the map

# Getting data from the files (currently only species names)

Abies_poly$species#gives the species name associated with "Abies_poly"

## End(Not run)
```

BIEN_ranges_intersect_species *Download range maps that intersect the range of a given species.*

Description

BIEN_ranges_intersect_species extracts range maps for a specified bounding box.

Usage

```
BIEN_ranges_intersect_species(
  species,
  directory = NULL,
  species.names.only = FALSE,
  include.focal = TRUE,
  return.species.list = TRUE,
  include.gid = FALSE,
  ...
)
```

Arguments

species	Focal species (or a vector of species) for which to extract intersecting ranges.
directory	Directory that range maps should be saved in. If none is specified, range maps will be saved in the current working directory.
species.names.only	Return species names rather than spatial data? Default is FALSE.
include.focal	Should a range for the focal species be downloaded? Default is TRUE.
return.species.list	Should a species list be returned in addition to downloading range maps? Default is FALSE
include.gid	Should the files returned have a unique GID appended to them? This is needed if downloading multiple maps for the same species.
...	Additional arguments passed to internal functions.

Value

Range maps for all available species that intersect the range of the focal species.

Note

Details on the construction of BIEN range maps is available at https://bien.nceas.ucsb.edu/bien/biendata/bien-3/

See Also

Other range functions: BIEN_ranges_box(), BIEN_ranges_genus(), BIEN_ranges_list(), BIEN_ranges_load_species
BIEN_ranges_sf(), BIEN_ranges_shapefile_to_skinny(), BIEN_ranges_skinny_ranges_to_richness_raster(),
BIEN_ranges_species(), BIEN_ranges_species_bulk()

Examples

```
## Not run:
temp_dir <- file.path(tempdir(), "BIEN_temp") #Set a working directory
BIEN_ranges_intersect_species(species = "Carnegiea_gigantea",
directory = temp_dir,include.focal = TRUE)
species_vector<-c("Carnegiea_gigantea","Echinocereus coccineus")
BIEN_ranges_intersect_species(species = species_vector,species.names.only = TRUE)

## End(Not run)
```

BIEN_ranges_list *List available range maps*

Description

BIEN_ranges_list a data.frame containing listing all range maps currently available.

Usage

```
BIEN_ranges_list(...)
```

Arguments

...	Additional arguments passed to internal functions.

Value

A data.frame containing the available species and their associated GIDs.

See Also

Other range functions: BIEN_ranges_box(), BIEN_ranges_genus(), BIEN_ranges_intersect_species(),
BIEN_ranges_load_species(), BIEN_ranges_sf(), BIEN_ranges_shapefile_to_skinny(), BIEN_ranges_skinny_ra
BIEN_ranges_species(), BIEN_ranges_species_bulk()

Other metadata functions: BIEN_metadata_citation(), BIEN_metadata_data_dictionaries(),
BIEN_metadata_database_version(), BIEN_metadata_list_political_names(), BIEN_metadata_match_data(),
BIEN_plot_metadata()

Examples

```
## Not run:
available_maps<-BIEN_ranges_list()
## End(Not run)
```

BIEN_ranges_load_species *Load range maps for specified species.*

Description

BIEN_ranges_load_species returns spatial data for the specified species.

Usage

```
BIEN_ranges_load_species(species, ...)
```

Arguments

species	A single species or a vector of species.
...	Additional arguments passed to internal functions.

Value

A sf containing range maps for the specified species.

See Also

Other range functions: BIEN_ranges_box(), BIEN_ranges_genus(), BIEN_ranges_intersect_species(), BIEN_ranges_list(), BIEN_ranges_sf(), BIEN_ranges_shapefile_to_skinny(), BIEN_ranges_skinny_ranges_to_r BIEN_ranges_species(), BIEN_ranges_species_bulk()

Examples

```
## Not run:
library(maps)
species_vector<-c("Abies_lasiocarpa","Abies_amabilis")
abies_maps <- BIEN_ranges_load_species(species = species_vector)
xanthium_strumarium <- BIEN_ranges_load_species(species = "Xanthium strumarium")

#Plotting files
plot(abies_maps) # plots the sf, but doesn't mean much without any reference
map('world', fill = TRUE, col = "grey")#plots a world map (WGS84 projection), in grey
plot(xanthium_strumarium,col="forest green",add = TRUE) #adds the range of X. strumarium
plot(abies_maps[1,], add = TRUE, col ="light green")

## End(Not run)
```

BIEN_ranges_sf *Download range maps that intersect a user-supplied sf object.*

Description

BIEN_ranges_sf extracts range maps that intersect a specified simple features (sf) object.

Usage

```
BIEN_ranges_sf(
  sf,
  directory = NULL,
  species.names.only = FALSE,
  return.species.list = TRUE,
  crop.ranges = FALSE,
  include.gid = FALSE,
  ...
)
```

Arguments

sf	An object of class sf.
directory	Directory that range maps should be saved in. If none is specified, range maps will be saved in the current working directory.
species.names.only	Return species names rather than spatial data? Default is FALSE.
return.species.list	Should a species list be returned in addition to downloading range maps? Default is FALSE
crop.ranges	Should the ranges be cropped to the focal area? Default is FALSE.
include.gid	Should the files returned have a unique GID appended to them? This is needed if downloading multiple maps for the same species.
...	Additional arguments passed to internal functions.

Value

All range maps that intersect the user-supplied sf object.

Note

Details on the construction of BIEN range maps is available at https://bien.nceas.ucsb.edu/bien/biendata/bien-3/

See Also

Other range functions: BIEN_ranges_box(), BIEN_ranges_genus(), BIEN_ranges_intersect_species(), BIEN_ranges_list(), BIEN_ranges_load_species(), BIEN_ranges_shapefile_to_skinny(), BIEN_ranges_skinny_ranges_to_richness_raster(), BIEN_ranges_species(), BIEN_ranges_species_bulk()

Examples

```
## Not run:

# Here we use a range map as our example polygon

BIEN_ranges_species("Carnegiea gigantea") #saves ranges to the current working directory

# Read in the polygon  with sf
sf <- sf::st_read(dsn = ".",
                  layer = "Carnegiea_gigantea")

BIEN_ranges_sf(sf = sf,
               limit = 10)
               # We use the limit argument to return only 10 range maps.
               # Omit the limit argument to get all ranges

#Note that this will save many shapefiles to the working directory.

## End(Not run)
```

BIEN_ranges_shapefile_to_skinny

Extract range data and convert to smaller "skinny" format

Description

BIEN_ranges_shapefile_to_skinny converts ranges to a "skinny" format to save space.

Usage

```
BIEN_ranges_shapefile_to_skinny(directory, raster, skinny_ranges_file = NULL)
```

Arguments

directory	The directory where range shapefiles will be stored. If NULL, a temporary directory will be used.
raster	A raster (which must have a CRS specified) to be used for rasterizing the ranges.
skinny_ranges_file	A filename that will be used to write the skinny ranges will be written to (RDS format). If NULL, this will not be written.

Value

Matrix containing 2 columns: 1) Species name; and 2) the raster cell number it occurs within.

See Also

Other range functions: BIEN_ranges_box(), BIEN_ranges_genus(), BIEN_ranges_intersect_species(), BIEN_ranges_list(), BIEN_ranges_load_species(), BIEN_ranges_sf(), BIEN_ranges_skinny_ranges_to_richness BIEN_ranges_species(), BIEN_ranges_species_bulk()

Examples

```
## Not run:
BIEN_ranges_shapefile_to_skinny(directory = BIEN_ranges_species_bulk(species = c("Acer rubrum")),
raster = terra::rast(crs = "+proj=laea +lat_0=15 +lon_0=-80 +x_0=0 +y_0=0 +datum=WGS84
+units=m +no_defs +ellps=WGS84 +towgs84=0,0,0",
           extent = terra::ext(c(-5261554,5038446,-7434988,7165012 )),
           resolution =c(100000,100000)
           )
)

## End(Not run)
```

BIEN_ranges_skinny_ranges_to_richness_raster

Build a richness raster from a skinny range file

Description

BIEN_ranges_skinny_ranges_to_richness_raster takes in "skinny" range data and converts it to a richness raster.

Usage

```
BIEN_ranges_skinny_ranges_to_richness_raster(skinny_ranges, raster)
```

Arguments

skinny_ranges	A matrix output by the function "BIEN_ranges_skinny" or equivalent methods.
raster	The raster that was used in building the skinny_ranges matrix.

Value

Raster

See Also

Other range functions: BIEN_ranges_box(), BIEN_ranges_genus(), BIEN_ranges_intersect_species(), BIEN_ranges_list(), BIEN_ranges_load_species(), BIEN_ranges_sf(), BIEN_ranges_shapefile_to_skinny(), BIEN_ranges_species(), BIEN_ranges_species_bulk()

Examples

```
## Not run:


template_raster <- terra::rast(
  crs = "+proj=laea +lat_0=15 +lon_0=-80 +x_0=0 +y_0=0 +datum=WGS84
  +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0",
  ext = ext(c(-5261554, 5038446, -7434988, 7165012 )),
  resolution =  c(100000, 100000))

#Download ranges and convert to a "skinny" format
skinny_ranges <- BIEN_ranges_shapefile_to_skinny(
directory = BIEN_ranges_species_bulk(species = c("Acer rubrum"),
raster = template_raster)

#Convert from skinny format to richness raster
richness_raster<- BIEN_ranges_skinny_ranges_to_richness_raster(
 skinny_ranges = skinny_ranges,raster = template_raster)

 plot(richness_raster)

## End(Not run)
```

BIEN_ranges_species *Download range maps for given species.*

Description

BIEN_ranges_species extracts range maps for the specified species.

Usage

```
BIEN_ranges_species(
  species,
  directory = NULL,
  matched = TRUE,
  match_names_only = FALSE,
  include.gid = FALSE,
  ...
)
```

Arguments

species	A single species or a vector of species.
directory	Directory that range maps should be saved in. If none is specified, range maps will be saved in the current working directory.
matched	Return a list of taxa that were downloaded. Default is TRUE.

match_names_only
: Check for range maps for the taxa specified without downloading range maps. Default is FALSE.

include.gid
: Should the files returned have a unique GID appended to them? This is needed if downloading multiple maps for the same species.

...
: Additional arguments passed to internal functions.

Value

Range maps for specified species.

Note

Details on the construction of BIEN range maps is available at https://bien.nceas.ucsb.edu/bien/biendata/bien-3/

See Also

Other range functions: BIEN_ranges_box(), BIEN_ranges_genus(), BIEN_ranges_intersect_species(), BIEN_ranges_list(), BIEN_ranges_load_species(), BIEN_ranges_sf(), BIEN_ranges_shapefile_to_skinny(), BIEN_ranges_skinny_ranges_to_richness_raster(), BIEN_ranges_species_bulk()

Examples

```
## Not run:
library(sf)
library(maps) #a convenient source of maps
species_vector <- c("Abies_lasiocarpa","Abies_amabilis")
BIEN_ranges_species(species_vector)
BIEN_ranges_species(species_vector, match_names_only = TRUE)
temp_dir <- file.path(tempdir(), "BIEN_temp")#Set a working directory
BIEN_ranges_species(species = species_vector,
                    directory = temp_dir)#saves ranges to a temporary directory
BIEN_ranges_species("Abies_lasiocarpa")
BIEN_ranges_species("Abies_lasiocarpa",
                    directory = temp_dir)

#Reading files

Abies_poly <- st_read(dsn = temp_dir,
                      layer = "Abies_lasiocarpa")

#Plotting files
plot(Abies_poly[1])#plots the range, but doesn't mean much without any reference
map('world', fill = TRUE, col = "grey")#plots a world map (WGS84 projection), in grey

plot(Abies_poly[1],
     col = "forest green",
     add = TRUE) #adds the range of Abies lasiocarpa to the map

# Getting data from the files (currently only species names and a BIEN ID field)
```

```
Abies_poly$species#gives the species name associated with "Abies_poly"
## End(Not run)#'
```

`BIEN_ranges_species_bulk` *Extract range data for large numbers of species*

Description

BIEN_ranges_species_bulk downloads ranges for a large number of species using parallel processing.

Usage

```
BIEN_ranges_species_bulk(
  species = NULL,
  directory = NULL,
  batch_size = 1000,
  return_directory = TRUE,
  use_parallel = FALSE
)
```

Arguments

`species`	A vector of species or NULL (the default). If NULL, all available ranges will be used.
`directory`	The directory where range shapefiles will be stored. If NULL, a temporary directory will be used.
`batch_size`	The number of ranges to download at once.
`return_directory`	Should the directory be returned? Default is TRUE
`use_parallel`	Logical. Should batches be downloaded in parallel? If set to TRUE, AND if parallel and foreach are available, parallel processing of downloads will use n-1 clusters.

Value

Optionally, the directory to which the files were saved.

Note

This function may take a long time (hours) to run depending on the number of cores, download speed, etc.

See Also

Other range functions: BIEN_ranges_box(), BIEN_ranges_genus(), BIEN_ranges_intersect_species(), BIEN_ranges_list(), BIEN_ranges_load_species(), BIEN_ranges_sf(), BIEN_ranges_shapefile_to_skinny(), BIEN_ranges_skinny_ranges_to_richness_raster(), BIEN_ranges_species()

Examples

```
## Not run:
#To download all BIEN ranges maps:
BIEN_ranges_species_bulk()

## End(Not run)
```

BIEN_stem_datasource *Extract stem data for a given datasource from BIEN*

Description

BIEN_stem_datasource downloads occurrence records for specific datasources from the BIEN database.

Usage

```
BIEN_stem_datasource(
  datasource,
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  political.boundaries = FALSE,
  collection.info = FALSE,
  all.metadata = FALSE,
  ...
)
```

Arguments

datasource	A single datasource, or a vector of datasources.
cultivated	Return known cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.
natives.only	Exclude detected introduced species? Default is TRUE.

`political.boundaries`
: Return information on political boundaries for an observation? The default value is FALSE.

`collection.info`
: Return additional information about collection and identification? The default value is FALSE.

`all.metadata`
: Should additional plot metadata be returned? Default is FALSE.

`...`
: Additional arguments passed to internal functions.

Value

Dataframe containing stem data for the specified datasource.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

Setting either "cultivated" or "native.status" to TRUE will significantly slow the speed of a query.

#' @note For a list of available datasources, use `BIEN_plot_list_datasource`.

See Also

Other stem functions: `BIEN_stem_family()`, `BIEN_stem_genus()`, `BIEN_stem_sampling_protocol()`, `BIEN_stem_species()`

Examples

```
## Not run:
BIEN_stem_datasource(datasource = "SALVIAS")
## End(Not run)
```

BIEN_stem_family *Extract stem data for specified families from BIEN*

Description

BIEN_stem_family downloads occurrence records for specific families from the BIEN database.

Usage

```
BIEN_stem_family(
  family,
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
```

```
  political.boundaries = FALSE,
  collection.info = FALSE,
  all.metadata = FALSE,
  ...
)
```

Arguments

family	A single family, or a vector of families. Families should be capitalized.
cultivated	Return known cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.
natives.only	Exclude detected introduced species? Default is TRUE.
political.boundaries	Return information on political boundaries for an observation? The default value is FALSE.
collection.info	Return additional information about collection and identification? The default value is FALSE.
all.metadata	Should additional plot metadata be returned? Default is FALSE.
...	Additional arguments passed to internal functions.

Value

Dataframe containing stem data for the specified families.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

Setting either "cultivated" or "native.status" to TRUE will significantly slow the speed of a query.

See Also

Other stem functions: BIEN_stem_datasource(), BIEN_stem_genus(), BIEN_stem_sampling_protocol(), BIEN_stem_species()

Examples

```
## Not run:
BIEN_stem_family(family = "Marantaceae")
family_vector<-c("Marantaceae", "Buxaceae")
BIEN_stem_family(family = family_vector)
BIEN_stem_family(family = family_vector, all.taxonomy = TRUE, native.status = TRUE)
## End(Not run)
```

BIEN_stem_genus *Extract stem data for specified genera from BIEN*

Description

BIEN_stem_genus downloads occurrence records for specific genera from the BIEN database.

Usage

```
BIEN_stem_genus(
  genus,
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  political.boundaries = FALSE,
  collection.info = FALSE,
  all.metadata = FALSE,
  ...
)
```

Arguments

genus	A single genus, or a vector of genera. Genera should be capitalized.
cultivated	Return known cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.
natives.only	Exclude detected introduced species? Default is TRUE.
political.boundaries	Return information on political boundaries for an observation? The default value is FALSE.
collection.info	Return additional information about collection and identification? The default value is FALSE.
all.metadata	Should additional plot metadata be returned? Default is FALSE.
...	Additional arguments passed to internal functions.

Value

Dataframe containing stem data for the specified genera.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

Setting either "cultivated" or "native.status" to TRUE will significantly slow the speed of a query.

See Also

Other stem functions: BIEN_stem_datasource(), BIEN_stem_family(), BIEN_stem_sampling_protocol(), BIEN_stem_species()

Examples

```
## Not run:
BIEN_stem_genus(genus = "Tovomita")
genus_vector<-c("Tovomita", "Myrcia")
BIEN_stem_genus(genus = genus_vector)
BIEN_stem_genus(genus = genus_vector, all.taxonomy = TRUE)
## End(Not run)
```

BIEN_stem_sampling_protocol

Download stem data using a specified sampling protocol.

Description

BIEN_stem_sampling_protocol downloads plot-based stem data using a specified sampling protocol.

Usage

```
BIEN_stem_sampling_protocol(
  sampling_protocol,
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  political.boundaries = FALSE,
  collection.info = FALSE,
  all.metadata = FALSE,
  ...
)
```

Arguments

sampling_protocol	A sampling protocol or vector of sampling protocols. See BIEN_plot_list_sampling_protocols for options.
cultivated	Return known cultivated records as well? Default is FALSE.
new.world	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
all.taxonomy	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
native.status	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.
natives.only	Exclude detected introduced species? Default is TRUE.
political.boundaries	Return information on political boundaries for an observation? The default value is FALSE.
collection.info	Return additional information about collection and identification? The default value is FALSE.
all.metadata	Should additional plot metadata be returned? Default is FALSE.
...	Additional arguments passed to internal functions.

Value

A dataframe containing all data from the specified sampling protocol.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

See Also

Other stem functions: BIEN_stem_datasource(), BIEN_stem_family(), BIEN_stem_genus(), BIEN_stem_species()

Examples

```
## Not run:
BIEN_stem_sampling_protocol("Point-intercept")
## End(Not run)
```

`BIEN_stem_species` *Extract stem data for specified species from BIEN*

Description

BIEN_stem_species downloads occurrence records for specific species from the BIEN database.

Usage

```
BIEN_stem_species(
  species,
  cultivated = FALSE,
  new.world = NULL,
  all.taxonomy = FALSE,
  native.status = FALSE,
  natives.only = TRUE,
  political.boundaries = FALSE,
  collection.info = FALSE,
  all.metadata = FALSE,
  ...
)
```

Arguments

`species`	A single species, or a vector of species. Genus and species should be separated by a space. Genus should be capitalized.
`cultivated`	Return known cultivated records as well? Default is FALSE.
`new.world`	NULL (The default) returns global records, TRUE returns only New World, and FALSE only Old World.
`all.taxonomy`	Return all taxonomic information? This includes the raw data as well as the "scrubbed" data.
`native.status`	Return information on introduction status? The default value is FALSE. A value of TRUE also returns additional information on introduction status.
`natives.only`	Exclude detected introduced species? Default is TRUE.
`political.boundaries`	Return information on political boundaries for an observation? The default value is FALSE.
`collection.info`	Return additional information about collection and identification? The default value is FALSE.
`all.metadata`	Should additional plot metadata be returned? Default is FALSE.
`...`	Additional arguments passed to internal functions.

Value

Dataframe containing stem data for the specified species.

Note

US FIA coordinates have been fuzzed and swapped, for more details see: https://www.fia.fs.fed.us/tools-data/spatial/Policy/index.php

Setting either "cultivated" or "native.status" to TRUE will significantly slow the speed of a query.

See Also

Other stem functions: BIEN_stem_datasource(), BIEN_stem_family(), BIEN_stem_genus(), BIEN_stem_sampling_protocol()

Examples

```
## Not run:
BIEN_stem_species("Abies amabilis")
species_vector<-c("Abies amabilis", "Acer nigrum")
BIEN_stem_species(species_vector)
BIEN_stem_species(species_vector,all.taxonomy = TRUE)
## End(Not run)
```

BIEN_taxonomy_family *Extract taxonomic information for families*

Description

BIEN_taxonomy_family downloads a dataframe of all taxonomic information for given families.

Usage

```
BIEN_taxonomy_family(family, ...)
```

Arguments

family	A single family or a vector of families.
...	Additional arguments passed to internal functions.

Value

Dataframe containing taxonomic information for the specified families.

See Also

Other taxonomy functions: BIEN_taxonomy_genus(), BIEN_taxonomy_species()

Examples

```
## Not run:
BIEN_taxonomy_family("Orchidaceae")
family_vector<-c("Orchidaceae","Poaceae")
BIEN_taxonomy_family(family_vector)
## End(Not run)
```

BIEN_taxonomy_genus *Extract taxonomic information for genera*

Description

BIEN_taxonomy_genus downloads a dataframe of all taxonomic information for given genera.

Usage

```
BIEN_taxonomy_genus(genus, ...)
```

Arguments

genus	A single genus or a vector of genera.
...	Additional arguments passed to internal functions.

Value

Dataframe containing taxonomic information for the specified genera.

See Also

Other taxonomy functions: BIEN_taxonomy_family(), BIEN_taxonomy_species()

Examples

```
## Not run:
BIEN_taxonomy_genus("Acer")
genus_vector<-c("Acer","Quercus")
BIEN_taxonomy_genus(genus_vector)
## End(Not run)
```

BIEN_taxonomy_species *Extract taxonomic information for species*

Description

BIEN_taxonomy_species downloads a dataframe of all taxonomic information for given species.

Usage

```
BIEN_taxonomy_species(species, ...)
```

Arguments

species	A single species or a vector of species.
...	Additional arguments passed to internal functions.

Value

Dataframe containing taxonomic information for the specified species.

See Also

Other taxonomy functions: BIEN_taxonomy_family(), BIEN_taxonomy_genus()

Examples

```
## Not run:
BIEN_taxonomy_species("Cannabis sativa")
species_vector<-c("Acer nigrum","Cannabis sativa")
BIEN_taxonomy_species(species_vector)
## End(Not run)
```

BIEN_trait_country *Download trait data for given country.*

Description

BIEN_trait_species extracts trait data for the species country.

Usage

```
BIEN_trait_country(
  country,
  trait.name = NULL,
  all.taxonomy = FALSE,
  political.boundaries = TRUE,
  source.citation = FALSE,
  ...
)
```

Arguments

country	A single country or a vector of countries.
trait.name	Optional. The trait or traits you want returned. If left blank, all traits will be returned.
all.taxonomy	Should full taxonomic information and TNRS output be returned? Default is FALSE.
political.boundaries	Should political boundary information (country, state, etc.) be returned? Default is FALSE.
source.citation	Should readable source information be downloaded for each record? Note that BIEN_metadata_citation may be more useful.
...	Additional arguments passed to internal functions.

Value

A dataframe of all available trait data for the given country.

See Also

Other trait functions: BIEN_trait_family(), BIEN_trait_genus(), BIEN_trait_list(), BIEN_trait_mean(), BIEN_trait_species(), BIEN_trait_trait(), BIEN_trait_traitbyfamily(), BIEN_trait_traitbygenus(), BIEN_trait_traitbyspecies(), BIEN_trait_traits_per_species()

Examples

```
## Not run:
BIEN_trait_country("South Africa")
BIEN_trait_country(country="South Africa",trait="whole plant growth form")
## End(Not run)
```

BIEN_trait_family *Download trait data for given families.*

Description

BIEN_trait_family extracts all trait data for the specified families.

Usage

```
BIEN_trait_family(
  family,
  all.taxonomy = FALSE,
  political.boundaries = FALSE,
  source.citation = FALSE,
  ...
)
```

Arguments

family	A single family or a vector of families.
all.taxonomy	Should full taxonomic information and TNRS output be returned? Default is FALSE.
political.boundaries	Should political boundary information (country, state, etc.) be returned? Default is FALSE.
source.citation	Should readable source information be downloaded for each record? Note that BIEN_metadata_citation may be more useful.
...	Additional arguments passed to internal functions.

Value

A dataframe of all data matching the specified families.

See Also

Other trait functions: BIEN_trait_country(), BIEN_trait_genus(), BIEN_trait_list(), BIEN_trait_mean(), BIEN_trait_species(), BIEN_trait_trait(), BIEN_trait_traitbyfamily(), BIEN_trait_traitbygenus(), BIEN_trait_traitbyspecies(), BIEN_trait_traits_per_species()

Examples

```
## Not run:
BIEN_trait_family("Poaceae")
family_vector<-c("Poaceae","Orchidaceae")
BIEN_trait_family(family_vector)
## End(Not run)
```

BIEN_trait_genus *Download trait data for given genera.*

Description

BIEN_trait_genus extracts entries that contain the specified genera.

Usage

```
BIEN_trait_genus(
  genus,
  all.taxonomy = FALSE,
  political.boundaries = FALSE,
  source.citation = FALSE,
  ...
)
```

Arguments

genus	A single genus or a vector of genera.
all.taxonomy	Should full taxonomic information and TNRS output be returned? Default is FALSE.
political.boundaries	Should political boundary information (country, state, etc.) be returned? Default is FALSE.
source.citation	Should readable source information be downloaded for each record? Note that BIEN_metadata_citation may be more useful.
...	Additional arguments passed to internal functions.

Value

A dataframe of all data matching the specified genera.

See Also

Other trait functions: BIEN_trait_country(), BIEN_trait_family(), BIEN_trait_list(), BIEN_trait_mean(), BIEN_trait_species(), BIEN_trait_trait(), BIEN_trait_traitbyfamily(), BIEN_trait_traitbygenus(), BIEN_trait_traitbyspecies(), BIEN_trait_traits_per_species()

Examples

```
## Not run:
BIEN_trait_genus("Acer")
genus_vector <- c("Acer","Abies")
BIEN_trait_genus(genus_vector)
## End(Not run)
```

BIEN_trait_list *List all available types of trait data*

Description

BIEN_trait_list produces a dataframe of all available types of trait data.

Usage

```
BIEN_trait_list(...)
```

Arguments

...	Additional arguments passed to internal functions.

Value

A dataframe containing all currently available types of trait data and details on measurement.

See Also

Other trait functions: BIEN_trait_country(), BIEN_trait_family(), BIEN_trait_genus(), BIEN_trait_mean(), BIEN_trait_species(), BIEN_trait_trait(), BIEN_trait_traitbyfamily(), BIEN_trait_traitbygenus(), BIEN_trait_traitbyspecies(), BIEN_trait_traits_per_species()

Examples

```
## Not run:
BIEN_trait_list()
## End(Not run)
```

BIEN_trait_mean *Calculates species mean values for a given trait, using Genus or Family level data where Species level data is lacking.*

Description

BIEN_trait_mean Estimates species mean values for a given trait, using Genus or Family level data where Species level data is absent.

Usage

```
BIEN_trait_mean(species, trait, ...)
```

Arguments

species	A single species or a vector of species.
trait	A single trait.
...	Additional arguments passed to internal functions.

Value

A dataframe of estimated trait means and associated metadata for the given species.

Note

Trait spelling needs to be exact and case-sensitive, see BIEN_trait_list for a list of traits.

See Also

Other trait functions: BIEN_trait_country(), BIEN_trait_family(), BIEN_trait_genus(), BIEN_trait_list(), BIEN_trait_species(), BIEN_trait_trait(), BIEN_trait_traitbyfamily(), BIEN_trait_traitbygenus(), BIEN_trait_traitbyspecies(), BIEN_trait_traits_per_species()

Examples

```
## Not run:
BIEN_trait_mean(species=c("Poa annua","Juncus trifidus"),trait="leaf dry mass per leaf fresh mass")
## End(Not run)
```

BIEN_trait_species *Download trait data for given species.*

Description

BIEN_trait_species extracts trait data for the species specified.

Usage

```
BIEN_trait_species(
  species,
  all.taxonomy = FALSE,
  political.boundaries = FALSE,
  source.citation = FALSE,
  ...
)
```

Arguments

species	A single species or a vector of species.
all.taxonomy	Should full taxonomic information and TNRS output be returned? Default is FALSE.
political.boundaries	Should political boundary information (country, state, etc.) be returned? Default is FALSE.
source.citation	Should readable source information be downloaded for each record? Note that BIEN_metadata_citation may be more useful.
...	Additional arguments passed to internal functions.

Value

A dataframe of all available trait data for the given species.

See Also

Other trait functions: BIEN_trait_country(), BIEN_trait_family(), BIEN_trait_genus(), BIEN_trait_list(), BIEN_trait_mean(), BIEN_trait_trait(), BIEN_trait_traitbyfamily(), BIEN_trait_traitbygenus(), BIEN_trait_traitbyspecies(), BIEN_trait_traits_per_species()

Examples

```
## Not run:
BIEN_trait_species("Poa annua")
species_vector<-c("Poa annua","Juncus trifidus")
BIEN_trait_species(species_vector)
## End(Not run)
```

BIEN_trait_trait *Download all measurements of a specific trait(s).*

Description

BIEN_trait_trait downloads all measurements of the trait(s) specified.

Usage

```
BIEN_trait_trait(
  trait,
  all.taxonomy = FALSE,
  political.boundaries = FALSE,
  source.citation = FALSE,
  ...
)
```

Arguments

trait	A single trait or a vector of traits.
all.taxonomy	Should full taxonomic information and TNRS output be returned? Default is FALSE.
political.boundaries	Should political boundary information (country, state, etc.) be returned? Default is FALSE.
source.citation	Should readable source information be downloaded for each record? Note that BIEN_metadata_citation may be more useful.
...	Additional arguments passed to internal functions.

Value

A dataframe of all available trait data for the given trait(s).

Note

Trait spelling needs to be exact and case-sensitive, see BIEN_trait_list for a list of traits.

See Also

Other trait functions: BIEN_trait_country(), BIEN_trait_family(), BIEN_trait_genus(), BIEN_trait_list(), BIEN_trait_mean(), BIEN_trait_species(), BIEN_trait_traitbyfamily(), BIEN_trait_traitbygenus(), BIEN_trait_traitbyspecies(), BIEN_trait_traits_per_species()

Examples

```
## Not run:
BIEN_trait_trait("whole plant height")
trait_vector<-c("whole plant height", "leaf dry mass per leaf fresh mass")
BIEN_trait_trait(trait_vector)
## End(Not run)
```

BIEN_trait_traitbyfamily

Download trait data for given families and traits.

Description

BIEN_trait_traitbyfamily extracts entries that contain the specified families and trait(s).

Usage

```
BIEN_trait_traitbyfamily(
  family,
  trait,
  all.taxonomy = FALSE,
  political.boundaries = FALSE,
  source.citation = FALSE,
  ...
)
```

Arguments

family	A single family or a vector of families.
trait	A single trait or a vector of traits.
all.taxonomy	Should full taxonomic information and TNRS output be returned? Default is FALSE.
political.boundaries	Should political boundary information (country, state, etc.) be returned? Default is FALSE.
source.citation	Should readable source information be downloaded for each record? Note that BIEN_metadata_citation may be more useful.
...	Additional arguments passed to internal functions.

Value

A dataframe of all data matching the specified trait(s) and family/families.

Note

Trait spelling needs to be exact and case-sensitive, see BIEN_trait_list for a list of traits.

See Also

Other trait functions: BIEN_trait_country(), BIEN_trait_family(), BIEN_trait_genus(), BIEN_trait_list(), BIEN_trait_mean(), BIEN_trait_species(), BIEN_trait_trait(), BIEN_trait_traitbygenus(BIEN_trait_traitbyspecies(), BIEN_trait_traits_per_species()

Examples

```
## Not run:
BIEN_trait_traitbyfamily(trait = "whole plant height", family = "Poaceae")
trait_vector <- c("whole plant height", "leaf fresh mass")
family_vector < -c("Orchidaceae","Poaceae")
BIEN_trait_traitbyfamily(trait = trait_vector, family = family_vector)
## End(Not run)
```

BIEN_trait_traitbygenus *Download trait data for given genus/genera and trait(s).*

Description

BIEN_trait_traitbygenus extracts entries that contain the specified genus/genera and trait(s).

Usage

```
BIEN_trait_traitbygenus(
  genus,
  trait,
  all.taxonomy = FALSE,
  political.boundaries = FALSE,
  source.citation = FALSE,
  ...
)
```

Arguments

genus	A single genus or a vector of genera.
trait	A single trait or a vector of traits.
all.taxonomy	Should full taxonomic information and TNRS output be returned? Default is FALSE.
political.boundaries	Should political boundary information (country, state, etc.) be returned? Default is FALSE.
source.citation	Should readable source information be downloaded for each record? Note that BIEN_metadata_citation may be more useful.
...	Additional arguments passed to internal functions.

Value

A dataframe of all data matching the specified trait(s) and genus/genera.

Note

Trait spelling needs to be exact and case-sensitive, see BIEN_trait_list for a list of traits.

See Also

Other trait functions: BIEN_trait_country(), BIEN_trait_family(), BIEN_trait_genus(), BIEN_trait_list(), BIEN_trait_mean(), BIEN_trait_species(), BIEN_trait_trait(), BIEN_trait_traitbyfamily, BIEN_trait_traitbyspecies(), BIEN_trait_traits_per_species()

Examples

```
## Not run:
BIEN_trait_traitbygenus(trait = "whole plant height", genus = "Carex")
trait_vector<-c("whole plant height", "leaf area")
genus_vector<-c("Carex","Betula")
BIEN_trait_traitbygenus(trait=trait_vector,genus=genus_vector)
## End(Not run)
```

BIEN_trait_traitbyspecies

Download trait data for given species and trait.

Description

BIEN_trait_traitbyspecies extracts entries that contain the specified species and trait(s).

Usage

```
BIEN_trait_traitbyspecies(
  species,
  trait,
  all.taxonomy = FALSE,
  political.boundaries = FALSE,
  source.citation = FALSE,
  ...
)
```

Arguments

species	A single species or a vector of species.
trait	A single trait or a vector of traits.
all.taxonomy	Should full taxonomic information and TNRS output be returned? Default is FALSE.
political.boundaries	Should political boundary information (country, state, etc.) be returned? Default is FALSE.
source.citation	Should readable source information be downloaded for each record? Note that BIEN_metadata_citation may be more useful.
...	Additional arguments passed to internal functions.

Value

A dataframe of all data matching the specified trait(s) and species.

Note

Trait spelling needs to be exact and case-sensitive, see BIEN_trait_list for a list of traits.

See Also

Other trait functions: BIEN_trait_country(), BIEN_trait_family(), BIEN_trait_genus(), BIEN_trait_list(), BIEN_trait_mean(), BIEN_trait_species(), BIEN_trait_trait(), BIEN_trait_traitbyfamily BIEN_trait_traitbygenus(), BIEN_trait_traits_per_species()

Examples

```
## Not run:
BIEN_trait_traitbyspecies(trait = "whole plant height", species = "Carex capitata")
trait_vector<-c("whole plant height", "leaf area")
species_vector<-c("Carex capitata","Betula nana")
BIEN_trait_traitbyspecies(trait=trait_vector,species=species_vector)
## End(Not run)
```

BIEN_trait_traits_per_species

Count the number of trait observations for each species in the BIEN database

---

Description

BIEN_trait_traits_per_species downloads a count of the number of records for each trait for each species in the BIEN database.

Usage

```
BIEN_trait_traits_per_species(species = NULL, ...)
```

Arguments

species	Optional species or vector of species. If left blank, returns counts for all species.
...	Additional arguments passed to internal functions.

Value

Returns a dataframe containing the number of trait records for each species in the BIEN database.

See Also

Other trait functions: BIEN_trait_country(), BIEN_trait_family(), BIEN_trait_genus(), BIEN_trait_list(), BIEN_trait_mean(), BIEN_trait_species(), BIEN_trait_trait(), BIEN_trait_traitbyfamily BIEN_trait_traitbygenus(), BIEN_trait_traitbyspecies()

Examples

```
## Not run:
trait_observation_counts<-BIEN_trait_traits_per_species()
## End(Not run)
```

Index