Package 'OpenRange'

October 28, 2026

Title Code to Access Open Access Species Range Maps

Version 0.0.3

Description Allows access to a proof-of-concept database containing Open Access species range models and relevant metadata. Access to the database is via both 'PostgreSQL' connection and API <https://github.com/EnquistLab/Biendata-Frontend>, allowing diverse use-cases.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0), RPostgreSQL

Imports DBI, sf, httr, jsonlite

Suggests knitr, rmarkdown, testthat (>= 3.0.0), maps, tidyverse, vcr, ggplot2

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Brian Maitner [aut, cre] (ORCID: <https://orcid.org/0000-0002-2118-9880>), Cory Merow [aut], Brad Boyle [aut], Xiao Feng [aut], Rethvick Sriram Yugendra Babu [aut], Brian Enquist [aut]

Maintainer Brian Maitner <bmaitner@usf.edu>

Repository CRAN

Date/Publication 2026-10-28 19:30:09 UTC

Contents

OpenRange_api_load_species 2
OpenRange_api_species 3

OpenRange_get_license 5
OpenRange_get_stats 6
OpenRange_list_scenarios 7
OpenRange_load_species 8
OpenRange_sf 9
OpenRange_species 11

Index **14**

`OpenRange_api_load_species` *Load range maps for given species using the API.*

Description

OpenRange_api_load_species extracts range maps for the specified species.

Usage

```
OpenRange_api_load_species(species, ...)
```

Arguments

`species`	A single species.
`...`	Additional arguments passed to internal functions.

Value

Range maps for specified species, or NULL if the API is unavailable or no maps are found.

Note

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

This function relies on an Internet resource. If that resource is unavailable, NULL will be returned along with an informative message.

See Also

Other range functions: `OpenRange_api_species()`, `OpenRange_get_license()`, `OpenRange_get_stats()`, `OpenRange_list_scenarios()`, `OpenRange_load_species()`, `OpenRange_sf()`, `OpenRange_species()`

Examples

```
library(maps) #a convenient source of maps
library(sf)

temp_dir <- file.path(tempdir(), "BIEN_temp")

#Download ranges
Abies_poly <- OpenRange_api_load_species(species = "Abies_amabilis")

# OpenRange functions return NULL if the data source is unavailable,
# so check the output before using it.

if(!is.null(Abies_poly)){

 #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,col="forest green",add=TRUE) #adds the range to the map

 #Getting data from the files
 Abies_poly |>
   st_drop_geometry()

}
```

`OpenRange_api_species` *Download range maps for given species using the API.*

Description

OpenRange_api_species downloads range maps for the specified species.

Usage

```
OpenRange_api_species(species, directory, include_id = TRUE, ...)
```

Arguments

`species`	A single species.
`directory`	Directory that range maps should be saved in.
`include_id`	Logical. Should the range_id be appended to the file name? Needed to save multiple maps per species.
`...`	Additional arguments passed to internal functions.

Value

NULL. Called for its side effect of downloading range maps.

Note

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

This function relies on an Internet resource. If that resource is unavailable, NULL will be returned along with an informative message.

See Also

Other range functions: OpenRange_api_load_species(), OpenRange_get_license(), OpenRange_get_stats(), OpenRange_list_scenarios(), OpenRange_load_species(), OpenRange_sf(), OpenRange_species()

Examples

```
library(OpenRange)
library(maps) #a convenient source of maps
library(sf)
library(ggplot2)


# Create temp directory

temp_dir <- file.path(tempdir(), "BIEN_temp")

#Download ranges

OpenRange_api_species(species = "Abies_amabilis",
                      directory = temp_dir,
                      include_id = TRUE) # saves ranges to a temporary directory

# Nothing is saved if the data source is unavailable, so check that the
# file exists before using it.

if(file.exists(file.path(temp_dir, "Abies_amabilis_117684.shp"))){

  #Reading files

  Abies_poly <- st_read(dsn = temp_dir,
                        layer = "Abies_amabilis_117684")

  # Get a map to plot on

  world <- map("world", plot = FALSE, fill = TRUE)|>
    st_as_sf()

  #Plotting files
```

```
  ggplot(data = world)+
    geom_sf()+
    geom_sf(data = Abies_poly,
            fill="green")+
    coord_sf(xlim = st_bbox(Abies_poly)[c(1,3)],
             ylim = st_bbox(Abies_poly)[c(2,4)]) +
    theme_bw()

  #Getting data from the files

  Abies_poly |>
    st_drop_geometry()

}
```

`OpenRange_get_license` *Download license information*

Description

OpenRange_get_license provides information on model licenses.

Usage

```
OpenRange_get_license(...)
```

Arguments

...	Additional arguments passed to internal functions.

Value

Data.frame containing information on the license associated with the maps, or NULL if the database is unavailable.

Note

This function relies on an Internet resource. If that resource is unavailable, NULL will be returned along with an informative message.

See Also

Other range functions: `OpenRange_api_load_species()`, `OpenRange_api_species()`, `OpenRange_get_stats()`, `OpenRange_list_scenarios()`, `OpenRange_load_species()`, `OpenRange_sf()`, `OpenRange_species()`

Other metadata functions: `OpenRange_get_stats()`, `OpenRange_list_scenarios()`

Examples

```
range_license_info <- OpenRange_get_license()

# OpenRange functions return NULL if the data source is unavailable,
# so check the output before using it.

if(!is.null(range_license_info)){

  head(range_license_info)

}
```

`OpenRange_get_stats` *Download model performance statistics*

Description

OpenRange_get_stats provides information on model performance.

Usage

```
OpenRange_get_stats(...)
```

Arguments

...	Additional arguments passed to internal functions.

Value

List object containing one data.frame for each of the three modeling frameworks, or NULL if the database is unavailable.

Note

This function relies on an Internet resource. If that resource is unavailable, NULL will be returned along with an informative message.

See Also

Other range functions: `OpenRange_api_load_species()`, `OpenRange_api_species()`, `OpenRange_get_license()`, `OpenRange_list_scenarios()`, `OpenRange_load_species()`, `OpenRange_sf()`, `OpenRange_species()`

Other metadata functions: `OpenRange_get_license()`, `OpenRange_list_scenarios()`

Examples

```
range_stats <- OpenRange_get_stats()

# OpenRange functions return NULL if the data source is unavailable,
# so check the output before using it.

if(!is.null(range_stats)){

  names(range_stats)

}
```

`OpenRange_list_scenarios` *Download information on available climate change scenarios*

Description

OpenRange_list_scenarios provides information on climate scenarios with range projections available.

Usage

```
OpenRange_list_scenarios(...)
```

Arguments

...	Additional arguments passed to internal functions.

Value

Data.frame containing climate change scenarios available in BIEN, or NULL if the database is unavailable.

Note

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

This function relies on an Internet resource. If that resource is unavailable, NULL will be returned along with an informative message.

See Also

Other range functions: `OpenRange_api_load_species()`, `OpenRange_api_species()`, `OpenRange_get_license()`, `OpenRange_get_stats()`, `OpenRange_load_species()`, `OpenRange_sf()`, `OpenRange_species()`

Other metadata functions: `OpenRange_get_license()`, `OpenRange_get_stats()`

Examples

```
## Not run:

cc_scenarios <- OpenRange_list_scenarios()


## End(Not run)
```

`OpenRange_load_species` *Load Open Range maps for specified species.*

Description

OpenRange_load_species returns spatial data for the specified species.

Usage

```
OpenRange_load_species(
  species,
  default_only = TRUE,
  projection = 4326,
  scenario = "present",
  ...
)
```

Arguments

`species`	A single species or a vector of species.
`default_only`	Logical. Should only default ranges be included? Default is TRUE.
`projection`	Numeric. What projection should maps be returned in? 4326 (default) or 3857
`scenario`	Which climate scenario(s) should be represented by maps? See BIEN_ranges_list_scenarios for options. Set to NULL to download all.
`...`	Additional arguments passed to internal functions.

Value

An sf object containing range maps for the specified species, or NULL if the database is unavailable or no maps are found.

Note

This function relies on an Internet resource. If that resource is unavailable, NULL will be returned along with an informative message.

See Also

Other range functions: OpenRange_api_load_species(), OpenRange_api_species(), OpenRange_get_license(), OpenRange_get_stats(), OpenRange_list_scenarios(), OpenRange_sf(), OpenRange_species()

Examples

```
#'
library(maps)
library(ggplot2)
library(sf)

species_vector <- c("Abies_lasiocarpa","Abies_amabilis")
abies_maps <- OpenRange_load_species(species = species_vector)

# OpenRange functions return NULL if the data source is unavailable,
# so check the output before using it.

if(!is.null(abies_maps)){

  # To get all maps for a species, use "default = FALSE".
  # Here, this returns maps with different thresholds from the same model
  std_all <- OpenRange_load_species(species = "Stellaria debilis",
                                    default_only = FALSE)

  #To just get the default map, use "default = TRUE"
  std_default <- OpenRange_load_species(species = "Stellaria debilis",
                                        default_only = TRUE)

  #get world map

  world <- map("world", plot = FALSE, fill = TRUE)|>
    st_as_sf()

  #Plotting ranges

  ggplot(data = world)+
    geom_sf()+
    geom_sf(data = abies_maps,
            mapping = aes(fill = species),
            alpha=0.5)+
    coord_sf(xlim = st_bbox(abies_maps)[c(1,3)],
             ylim = st_bbox(abies_maps)[c(2,4)]) +
    theme_bw()

}
```

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

Description

OpenRange_sf extracts range maps that intersect a specified SpatialPolygons or SpatialPolygonsDataFrame object.

Usage

```
OpenRange_sf(
  sf,
  directory,
  species.names.only = FALSE,
  return.species.list = TRUE,
  crop.ranges = FALSE,
  include.gid = FALSE,
  projection = 4326,
  scenario = "present",
  default_only = TRUE,
  ...
)
```

Arguments

sf	An object of class sf
directory	Directory that range maps should be saved in.
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.
projection	Numeric. What projection should maps be returned in? 4326 (default) or 3857
scenario	Which climate scenario should be represented by maps? See BIEN_ranges_list_scenarios.
default_only	Logical. Should only default ranges be included? Default is TRUE.
...	Additional arguments passed to internal functions.

Value

All range maps that intersect the user-supplied shapefile, or NULL if the database is unavailable.

Note

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

This function relies on an Internet resource. If that resource is unavailable, NULL will be returned along with an informative message.

See Also

Other range functions: OpenRange_api_load_species(), OpenRange_api_species(), OpenRange_get_license(), OpenRange_get_stats(), OpenRange_list_scenarios(), OpenRange_load_species(), OpenRange_species()

Examples

```
saguaro_poly <- OpenRange_load_species("Carnegiea gigantea")

# OpenRange functions return NULL if the data source is unavailable,
# so check the output before using it.

if(!is.null(saguaro_poly)){

 #Create a temp directory
 temp_dir <- file.path(tempdir(), "BIEN_temp")

 #Get range maps for all species with ranges that overlap the range of saguaros
 OpenRange_sf(sf = saguaro_poly,
              directory = temp_dir)

 #Note that this will save many sfs to the directory (or working directory)

}
```

OpenRange_species *Download range maps for given species.*

Description

OpenRange_species extracts range maps for the specified species.

Usage

```
OpenRange_species(
  species,
  directory,
  default_only = TRUE,
  matched = TRUE,
  match_names_only = FALSE,
  include_id = TRUE,
  projection = 4326,
  scenario = "present",
  ...
)
```

Arguments

species	A single species or a vector of species.
directory	Directory that range maps should be saved in.
default_only	Logical. Should only default ranges be included? Default is TRUE.
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_id	Logical. Should the range_id be appended to the file name? Needed to save multiple maps per species.
projection	Numeric. What projection should maps be returned in? 4326 (default) or 3857
scenario	Which climate scenario should be represented by maps? See BIEN_ranges_list_scenarios. Set to NULL to download all.
...	Additional arguments passed to internal functions.

Value

Range maps for specified species, or NULL if the database is unavailable.

Note

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

This function relies on an Internet resource. If that resource is unavailable, NULL will be returned along with an informative message.

See Also

Other range functions: OpenRange_api_load_species(), OpenRange_api_species(), OpenRange_get_license(), OpenRange_get_stats(), OpenRange_list_scenarios(), OpenRange_load_species(), OpenRange_sf()

Examples

```
library(maps) #a convenient source of maps
library(sf)

species_vector <- c("Abies_lasiocarpa","Abies_amabilis")

#check whether the species are available
OpenRange_species(species_vector,match_names_only = TRUE)

#Create a temp directory
temp_dir <- file.path(tempdir(), "BIEN_temp")

#Download ranges
OpenRange_species(species = species_vector,
```

```
                directory = temp_dir)#saves ranges to a temporary directory

# Nothing is saved if the data source is unavailable, so check that the
# file exists before using it.

if(file.exists(file.path(temp_dir, "Abies_amabilis_117684.shp"))){

 #Reading files

 Abies_poly <- st_read(dsn = temp_dir,
                       layer = "Abies_amabilis_117684")

 #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,col="forest green",add=TRUE) #adds the range of Abies lasiocarpa to the map

 #Getting data from the files
 Abies_poly |>
   st_drop_geometry()

}
```

Index

∗ **metadata functions**
OpenRange_get_license, 5
OpenRange_get_stats, 6
OpenRange_list_scenarios, 7
∗ **range functions**
OpenRange_api_load_species, 2
OpenRange_api_species, 3
OpenRange_get_license, 5
OpenRange_get_stats, 6
OpenRange_list_scenarios, 7
OpenRange_load_species, 8
OpenRange_sf, 9
OpenRange_species, 11

OpenRange_api_load_species, 2, *4–7*, *9*, *11*, *12*
OpenRange_api_species, *2*, 3, *5–7*, *9*, *11*, *12*
OpenRange_get_license, *2*, *4*, 5, *6*, *7*, *9*, *11*, *12*
OpenRange_get_stats, *2*, *4*, *5*, 6, *7*, *9*, *11*, *12*
OpenRange_list_scenarios, *2*, *4–6*, 7, *9*, *11*, *12*
OpenRange_load_species, *2*, *4–7*, 8, *11*, *12*
OpenRange_sf, *2*, *4–7*, *9*, 9, *12*
OpenRange_species, *2*, *4–7*, *9*, *11*, 11