

Package ‘bRacatus’

December 21, 2022

Title A Method to Estimate the Accuracy and Biogeographical Status of Georeferenced Biological Data

Version 1.0.11

Language en-GB

Description Automated assessment of accuracy and geographical status of georeferenced biological data. The methods rely on reference regions, namely checklists and range maps. Includes functions to obtain data from the Global Biodiversity Information Facility <<https://www.gbif.org/>> and from the Global Inventory of Floras and Traits <<https://gift.uni-goettingen.de/home>>. Alternatively, the user can input their own data. Furthermore, provides easy visualisation of the data and the results through the plotting functions. Especially suited for large datasets. The reference for the methodology is: Arlé et al. (under review).

Depends R (>= 3.5.0)

Imports data.table, geojsonio, graphics, grDevices, jsonlite, mapproj, methods, plotfunctions, raster, rgbif, rgdal, rgeos, rworldmap, sp, stats

License GPL (>= 2)

Encoding UTF-8

URL <https://github.com/EduardoArle/bRacatus>

BugReports <https://github.com/EduardoArle/bRacatus/issues>

LazyData true

RoxygenNote 7.2.3

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Eduardo Arlé [aut, cre],
Alexander Zizka [aut],
Patrick Weigelt [ctb],
Sam Levin [ctb],
Carsten Meyer [ths]

Maintainer Eduardo Arlé <eduardoarle@tauex.tau.ac.il>

Repository CRAN

Date/Publication 2022-12-21 10:10:02 UTC

R topics documented:

accuracy	2
availableCountries	3
biogeoStatus	3
countryChecklist	4
getOcc	4
giftRegions	5
giveOcc	5
giveRegions	6
glonafRegions	7
H_mirandae_sp	8
ID_raster	8
Model_accuracy	8
Model_biogeo	9
occSpatialPoints	9
plotAccuracy	10
plotBiogeoStatus	11
plotOcc	12
plotRefReg	13
pts	13
pts2	14
rangeMaps	14
Range_Phalanger_orientalis	15
signalCalculation	15
signals	16
signals_2	16
signals_3	16
Index	17

accuracy

accuracy

Description

Estimates the Accuracy of Each Point Record.

Usage

accuracy(signals)

Arguments

signals output of the function "signalCalculation". A data.frame including the original point data and the signals sent by the reference regions.

Value

The data.frame with the species occurrence information and an extra column indicating the estimated accuracy of each point.

availableCountries *availableCountries*

Description

List of countries and entities names for checklists

Usage

```
availableCountries()
```

Value

This function provides a list of countries and entities names available with rworldmaps for checklists

Examples

```
country_list <- availableCountries()
```

biogeoStatus *biogeoStatus*

Description

Estimates the biogeographic status of each point record.

Usage

```
biogeoStatus(signals)
```

Arguments

signals output of the function signalCalculation. A dataFrame including the original point data and the signals sent by the reference regions.

Value

The dataFrame with the species occurrence information and an extra column indicating the estimated biogeographic status of each point.

countryChecklist	<i>countryChecklists</i>
------------------	--------------------------

Description

Prepares user provided reference regions on a country level

Usage

```
countryChecklist(countries, biogeo_status)
```

Arguments

`countries` vector with one or more country names
`biogeo_status` vector informing the status of each country: alien, native or unknown

Value

This function provides shapefiles of countries with the correspondent biogeographic status of the species.

Examples

```
country_checklist <- countryChecklist(
  c("Brazil", "Argentina", "Uruguay", "Paraguay"),
  c("native", "alien", "unknown", "native"))
```

getOcc	<i>getOcc</i>
--------	---------------

Description

Downloads GBIF records iterating when necessary to overcome the limitation of 200,000 records

Usage

```
getOcc(species)
```

Arguments

`species` character, species binomial name

Value

This function downloads all records for a species from GBIF that have coordinates info. If necessary it loops several times to overcome the limit of 200,000 occurrences imposed by `occ_search` function. It returns a data table.

Examples

```
sps_occurrence <- getOcc("Babiana tubulosa")
```

giftRegions	<i>giftRegions</i>
-------------	--------------------

Description

Gets regions listed by GIFT for plant species

Usage

```
giftRegions(species, min_size = 1000, max_size = 1e+11)
```

Arguments

species	character, species binomial name
min_size	numeric, minimum size of checklists (in km2) to be included in the analysis.
max_size	numeric, maximum size of checklists (in km2) to be included in the analysis.

Value

This function returns a list containing three shapefiles derived by information supplied by GIFT. "regs" includes all the features corresponding to regions where the species has been listed as present. "regs_native" includes all the features corresponding to regions where the species has been listed as native. And "regs_alien" includes all the features corresponding to regions where the species has been listed as alien.

Examples

```
gift_reference_regions <- giftRegions("Babiana tubulosa")
```

giveOcc	<i>Input occurrence data</i>
---------	------------------------------

Description

Prepares user provided georeferenced biological data for the models

Usage

```
giveOcc(
  occ_data,
  species = "species",
  longitude = "longitude",
  latitude = "latitude"
)
```

Arguments

occ_data	table containing latitude and longitude
species	col.name containing the species information
longitude	col.name containing the longitude information
latitude	col.name containing the latitude information

Value

This function standardises the user provided georeferenced biological data to be fed into the models.

Examples

```
# Create a data.frame containing species names and coordinates

test_data <- data.frame(sps=rep("Equus acephalus",10),
  lon=c(-43.2, -58.4, -56, -44, -54.5, -57.4, -60.1, -68.5, -71.3, -47.5),
  lat=c(-22.9, -34.6, -34.8, -20, -25.5, -25.2, -3, -32.5, -41.1, -15.5),
  gender=rep("female",10), head_size=rep("headless individual"))

sps_occurrence <- giveOcc(test_data, "sps", "lon", "lat")
```

giveRegions

giveRegions

Description

Input checklist regions

Usage

```
giveRegions(regs, regs_native, regs_alien)
```

Arguments

regs	shapefile containing all regions of occurrence.
regs_native	shapefile containing regions where the species is native.
regs_alien	shapefile containing regions where the species is alien.

Value

This function returns a list containing three shapefiles derived by information supplied by GIFT. "regs" includes all the features corresponding to regions where the species has been listed as present. "regs_native" includes all the features corresponding to regions where the species has been listed as native. And "regs_alien" includes all the features corresponding to regions where the species has been listed as alien.

Examples

```
library(rworldmap)
world <- getMap()
regs <- world[c("Brazil", "Argentina", "Uruguay", "Paraguay"),]
regs_native <- world[c("Brazil", "Paraguay"),]
regs_alien <- world[c("Argentina"),]
regs_list <- giveRegions(regs, regs_native, regs_alien)
```

glonafRegions

glonafRegions

Description

Gets regions listed by GloNAF for plant species

Usage

```
glonafRegions(species, native = "gift", nat_ref_reg = NULL)
```

Arguments

species	character, species binomial name
native	character, source for the native reference regions. Options are "gift", "range map", or "checklist". If "gift" is chosen, the function will automatically download native regions listed by GIFT for the species. If "range map" or "checklist" is chosen, the user must provide a shapefile with either the species range map, or the features representing regions where it has been listed as native. Default is "gift".
nat_ref_reg	shapefile containing either the species native range map or checklist. The user must inform which reference region data type is being provided in the parameter "native".

Value

This function returns a list containing three shapefiles derived by information supplied by GloNAF for the alien reference regions, and the chosen source for the native reference regions. "regs" includes all the features corresponding to regions where the species has been listed as present. "regs_native" includes all the features corresponding to regions where the species has been listed as native. And "regs_alien" includes all the features corresponding to regions where the species has been listed as alien.

Examples

```
glonaf_reference_regions <- glonafRegions("Ambrosia grayi")
```

H_mirandae_sp	<i>Hemitriccus mirandae spatialPoints exemplary file</i>
---------------	--

Description

A SpatialPointsDataFrame containing the occurrences of Hemitriccus mirandae downloaded from GBIF

Examples

```
data("H_mirandae_sp")
```

ID_raster	<i>Cell ID raster</i>
-----------	-----------------------

Description

A Raster half-degree raster of the world with unique IDs per cell

Examples

```
data("ID_raster")
```

Model_accuracy	<i>Accuracy model</i>
----------------	-----------------------

Description

A glm accuracy model

Examples

```
data("Model_accuracy")
```

Model_biogeo	<i>Biogeographical model</i>
--------------	------------------------------

Description

A glm biogeographical model

Examples

```
data("Model_biogeo")
```

occSpatialPoints	<i>occSpatialPoints</i>
------------------	-------------------------

Description

Downloads gbif records iterating when necessary to overcome the limitation of 200,000 records

Usage

```
occSpatialPoints(occ)
```

Arguments

occ	table
-----	-------

Value

This function creates spatialPoints from tables containing coordinates.

Examples

```
# Create a data.frame containing species names and coordinates

test_data <- data.frame(sps=rep("Equus acephalus",10),
  lon=c(-43.2, -58.4, -56, -44, -54.5, -57.4, -60.1, -68.5, -71.3, -47.5),
  lat=c(-22.9, -34.6, -34.8, -20, -25.5, -25.2, -3, -32.5, -41.1, -15.5),
  gender=rep("female",10), head_size=rep("headless individual"))

sps_occurrence <- giveOcc(test_data, "sps", "lon", "lat")

sps_sp <- occSpatialPoints(sps_occurrence)
```

`plotAccuracy`*plotAccuracy*

Description

Plot the species occurrences showing the estimated accuracy of points.

Usage

```
plotAccuracy(  
  acc,  
  regional = TRUE,  
  reg.by = "country",  
  borders = TRUE,  
  col.features = "khaki",  
  col.bg = "azure2",  
  plot.range = FALSE,  
  range = NULL,  
  box = FALSE  
)
```

Arguments

<code>acc</code>	dataTable of the species occurrence including a column with the estimated accuracy of points.
<code>regional</code>	logical, whether the whole world should be plotted as the background or only the region adjacent to the species countries of occurrence.
<code>reg.by</code>	character, by countries where there are points or by area where the points are located.
<code>borders</code>	logical, whether country limits should be plotted.
<code>col.features</code>	colour for plotting features.
<code>col.bg</code>	colour for plotting the background.
<code>plot.range</code>	logical, if TRUE, range maps should be provided as a shapefile in argument <code>range</code> .
<code>range</code>	shapefile, species range map.
<code>box</code>	logical, includes frame with coordinates locations.

Value

This function plots the species occurrence with estimated accuracy of all points.

plotBiogeoStatus *plotBiogeoStatus*

Description

Plot the species occurrences showing the estimated biogeographical status of points.

Usage

```
plotBiogeoStatus(  
  biogeo,  
  regional = TRUE,  
  reg.by = "country",  
  borders = TRUE,  
  col.features = "khaki",  
  col.bg = "azure2",  
  plot.range = FALSE,  
  range = NULL,  
  box = FALSE  
)
```

Arguments

biogeo	dataTable of the species occurrence including a column with the estimated biogeographical status of points.
regional	logical, whether the whole world should be plotted as the background or only the region adjacent to the species countries of occurrence.
reg.by	character, by countries where there are points or by area where the points are located.
borders	logical, whether country limits should be plotted.
col.features	colour for plotting features.
col.bg	colour for plotting the background.
plot.range	logical, if TRUE, range maps should be provided as a shapefile in argument range.
range	shapefile, species range map.
box	logical, includes frame with coordinates locations.

Value

This function plots the species occurrence with estimated biogeographical status of all points.

`plotOcc`*plotOcc*

Description

Plot the species occurrences with map background for visualisation

Usage

```
plotOcc(occ, regional = TRUE)
```

Arguments

<code>occ</code>	dataTable of the species occurrence.
<code>regional</code>	logical, whether the whole world should be plotted as the background or only the region adjacent to the species countries of occurrence.

Value

This function plots the species occurrence

Examples

```
occ <- getOcc("Hemitriccus mirandae")

plotOcc(occ)

test_data <- data.frame(sps=rep("Equus acephalus",10),
  lon=c(-43.2, -58.4, -56, -44, -54.5, -57.4, -60.1, -68.5, -71.3, -47.5),
  lat=c(-22.9, -34.6, -34.8, -20, -25.5, -25.2, -3, -32.5, -41.1, -15.5),
  gender=rep("female",10), head_size=rep("headless individual"))

occ <- giveOcc(test_data, "sps", "lon", "lat")

plotOcc(occ)
# Plot occurrences with the whole world as background

plotOcc(occ, regional=FALSE)
```

`plotRefReg`*plotRefReg*

Description

Plot the species reference regions with map background for visualisation

Usage

```
plotRefReg(ref_reg)
```

Arguments

`ref_reg` list containing three shapefiles derived by information supplied by GIFT. "regs" includes all the features corresponding to regions where the species has been listed as present. "regs_native" includes all the features corresponding to regions where the species has been listed as native. And "regs_alien" includes all the features corresponding to regions where the species has been listed as alien..

Value

This function plots three maps of the species occurrence, showing the regions where it is present, native and alien.

`pts`*pts example 1*

Description

A DataFrame pts example 1

Examples

```
data("pts")
```

pts2

*pts example 2***Description**

A DataFrame pts example 2

Examples

```
data("pts2")
```

rangeMaps

*rangeMaps***Description**

Prepares range maps input by the user to be used as reference regions

Usage

```
rangeMaps(
  range,
  biogeo = "legend",
  native = "Extant (resident)",
  alien = "Introduced"
)
```

Arguments

range	SpatialPolygonsDataFrame
biogeo	character, name of the column containing information on biogeographic status of features
native	character, entries in biogeo column representing the native range of the species
alien	character, entries in biogeo column representing the alien range of the species

Value

This function returns a list containing three shapefiles derived from information supplied by the species range map in a shapefile format. "regs" includes all the features corresponding to regions where the species has been listed as present. "regs_native" includes all the features corresponding to regions where the species has been listed as native. And "regs_alien" includes all the features corresponding to regions where the species has been listed as alien.

Examples

```
range_map_reference_regions <- rangeMaps(Range_Phalanger_orientalis)
```

```
Range_Phalanger_orientalis
      Range Phalanger orientalis
```

Description

A SpatialPolygonsDataFrame Range Phalanger orientalis

Examples

```
data("Range_Phalanger_orientalis")
```

```
signalCalculation      signalCalculation
```

Description

Calculates signals sent from reference regions to point records.

Usage

```
signalCalculation(ref_reg, pts, biogeo = TRUE)
```

Arguments

ref_reg	a list of shapefiles containing checklist regions as "presence", "native", and "alien" categories. These can be original checklists, or checklists derived from range maps.
pts	data.frame containing the point records and their coordinates.
biogeo	logical, whether the biogeographical status indices should be calculated or not. Default is true, however at least the native reference regions must be included in the data.

Value

The data.frame of species occurrences with extra columns containing the location ID and presence signals for each point. If biogeo=TRUE, the data.frame also includes the nativeness and alienness indices.

`signals`*signals example 1*

Description

A SpatialPolygonsDataFrame signals example 1

Examples

```
data("signals")
```

`signals_2`*signals example 2*

Description

A SpatialPolygonsDataFrame signals example 2

Examples

```
data("signals_2")
```

`signals_3`*signals example 3*

Description

A SpatialPolygonsDataFrame signals example 3

Examples

```
data("signals_3")
```


Index

* **exemplary**

- H_mirandae_sp, 8
- pts, 13
- pts2, 14
- Range_Phalanger_orientalis, 15
- signals, 16
- signals_2, 16
- signals_3, 16

* **model**

- Model_accuracy, 8
- Model_biogeo, 9

* **object**

- H_mirandae_sp, 8
- Model_accuracy, 8
- Model_biogeo, 9
- pts, 13
- pts2, 14
- Range_Phalanger_orientalis, 15
- signals, 16
- signals_2, 16
- signals_3, 16

* **reference**

- ID_raster, 8

accuracy, 2

availableCountries, 3

biogeoStatus, 3

countryChecklist, 4

getOcc, 4

giftRegions, 5

giveOcc, 5

giveRegions, 6

glonafRegions, 7

H_mirandae_sp, 8

ID_raster, 8

Model_accuracy, 8

Model_biogeo, 9

occSpatialPoints, 9

plotAccuracy, 10

plotBiogeoStatus, 11

plotOcc, 12

plotRefReg, 13

pts, 13

pts2, 14

Range_Phalanger_orientalis, 15

rangeMaps, 14

signalCalculation, 15

signals, 16

signals_2, 16

signals_3, 16