Package ‘speciesgeocodeR’

October 17, 2015

Type Package

Title Prepare Species Distributions for the Use in Phylogenetic Analyses

Version 1.0-4

Date 2015-10-17

Author Alexander Zizka [aut, cre]

Maintainer Alexander Zizka <alexander.zizka@bioenv.gu.se>

Description Preparation of species occurrences and distribution data for the use in phylogenetic analyses. SpeciesgeocodeR is built for data cleaning, data exploration and data analysis and especially suited for biogeographical and ecological questions on large datasets. The package includes the easy creation of summary-tables and -graphs and geographical maps, the automatic cleaning of geographic occurrence data, the calculating of coexistence matrices and species ranges (EOO) as well as mapping diversity in geographic areas.

License GPL-3

Depends R (>= 3.0.0), maps

Imports stats, methods, utils, grDevices, graphics, sp, raster

Suggests ape, geiger, geosphere, rgbif, maptools

LazyData true

NeedsCompilation no

R topics documented:

`speciesgeocodeR-package`

Prepare Species Distributions for the Use in Phylogenetic Analyses

---

Description

Preparation of species occurrences and distribution data for the use in phylogenetic analyses. SpeciesgeocodeR is built for data cleaning, data exploration and data analysis and especially suited for biogeographical and ecological questions on large datasets. The package includes the easy creation of summary-tables and -graphs and geographical maps, the automatic cleaning of geographic occurrence data, the calculating of coexistence matrices and species ranges (EOO) as well as mapping diversity in geographic areas.

Details

Package:	speciesgeocodeR
Type:	Package
Version:	1.04
Date:	2015-10-17
License:	GPL-3

The core functions of the package can be used via the wrapper function `SpeciesGeoCoder`, with two input text files and one single command. In addition to this the package includes a set of functions to visualize geographic distribution data and diversity.

Author(s)

Alexander Zizka

Maintainer: Alexander Zizka, alexander.zizka@bioenv.gu.se

Examples

```
## Not run:
data(lemurs)
data(mdg_poly)

SpeciesGeoCoder(lemurs, mdg_poly)
conv <- Spgc2Biogeobears(outp)

## End(Not run)

data(lemurs)
data(mdg_poly)
inp <- ReadPoints(lemurs, mdg_poly)

outp <- SpGeoCodH(inp)
e <- c(42, 52, -27, -10)
ivesia_abu <- RichnessGrid(outp, e, reso = 60, type = "abu")

outp <- SpGeoCodH(inp)
e <- c(42, 52, -27, -10)
lemurs_div <- RichnessGrid(outp, e, reso = 60, type = "spnum")
MapGrid(lemurs_div)
```

CalcRange *Calculate the Extent of Occurrence*

Description

Uses the areaPolygon function of the geosphere package to calculate the extent of occurrence (EOO) for each species in the list, which can be used for criterion B of the IUCN red list classification.

Usage

```
CalcRange(x, mode =  "EOO", value = c("area", "shape"))
```

Arguments

x	data.frame, with 3 columns: 1 = species name, 2 = longitude coordinates, y = latitude coordinates.
mode	character string, setting the mode of calculation, currently only “EOO” is supported.
value	character string, setting the output value. If value = “area”: A data.frame with the EOO for each species in square kilometers; if value = “shape”, a list of convex hull polygons.

Value

If value = “area”: A data.frame with the EOO for each species in square kilometers; if value = “shape”, a list of convex hull polygons.

Note

See the speciesgeocodeR documentation for further information and examples.

Examples

```
data(lemurs)
CalcRange(lemurs)
```

CoExClass *Species Coexistence Matrices in Given Areas.*

Description

Add a coexistence matrix to an object of the class spgeoOUT created by SpGeoCod.

Usage

```
CoExClass(x)
```

Arguments

x	An object of the class spgeoOUT, where the coexistence matrix shall be added

Value

An object of the class spgeoOUT, including a coexistence matrix which can be addressed with via the “coexistence_classified” slot. The matrix will have the same dimensions as there are input species and will give the percent of occurrence points of each species (row) with all other species (columns) given the input polygons.

Note

See the speciesgeocodeR tutorial for further information and examples.

Examples

```
data(lemurs)
data(mdg_poly)

inp <- ReadPoints(lemurs, mdg_poly)
outp <- SpGeoCodH(inp)
outpcoex <- CoExClass(outp)
outpcoex$coexistence_classified
```

countryref *Country Centroids and Country Capitals*

Description

A `data.frame` with coordinates of country centroids and country capitals as reference for the `GeoClean` function. Coordinates are based on the Central Intelligence Agency World Factbook as provided at http://opengeocode.org/download/cow.php.

Usage

```
data("countryref")
```

Format

A data frame with 249 observations on 7 variables.

Source

CENTRAL INTELLIGENCE AGENCY (2014) *The World Factbook*, Washington, DC.

`http://opengeocode.org/download/cow.php`

Examples

```
data(countryref)
```

GeoClean *Automated Cleaning of Geographic Coordinates*

Description

Provides a number of different tests to clean datasets with geographic coordinates. Each function argument represents a different cleaning step.

Usage

```
GeoClean(x, isna = TRUE, isnumeric = TRUE,
         coordinatevalidity = TRUE, containszero = TRUE,
 zerozero = TRUE, zerozerothresh = 1,
 latequallong = TRUE, GBIFhead = FALSE,
 countrycentroid = FALSE, contthresh = 0.5,
 capitalcoords = FALSE, capthresh = 0.5,
 countrycheck = FALSE, polygons,
 referencecountries= countryref,
 outp = c("summary", "detailed", "cleaned"))
```

Arguments

`x` a `data.frame` with at least three columns: “identifier” (species name), “XCOOR” (longitude) and “YCOOR” (latitude). Column names might also be “species”, “longitude” and “latitude”. If the arguments “countrycentroid”, “capitalcoords” or “countrycheck” should be used, a fourth column named “country” is needed with the country names in ISO2 or ISO3. Alternatively, a `data.frame` as downloaded from GBIF.

`isna` logical. If TRUE, checks for missing values in the coordinates. Default = TRUE.

`isnumeric` logical. If TRUE, checks for non-numeric values in the coordinates. Default = TRUE.

`coordinatevalidity` logical. If TRUE, checks for non-valid coordinates (XCOOR > 180 and < -180; YCOOR >90 and <-90). Default = TRUE.

`containszero` logical. If TRUE, checks for coordinates that are exactly zero. Default = TRUE.

`zerozero` logical. If TRUE, checks if the coordinate fall within a rectangle around the point 0/0. Default = TRUE.

`zerozerothresh` numeric. The size of the rectangle around 0/0 in decimal degrees. Default = 0.5.

`latequallong` logical. If TRUE, checks for rows where XCOOR = YCOOR. Default = TRUE.

`GBIFhead` logical. If TRUE, checks if the coordinate fall within a 0.5 degree rectangle around the GBIF headquarters in Copenhagen. Default = FALSE.

`countrycentroid` logical. If TRUE checks if the coordinate fall within a rectangle around the centroid of the country specified in x$country. The size of the rectangle can be controlled using the "countthresh" argument. Default = FALSE.

`contthresh` numeric. The size of the rectangle around the country centroid (in degrees). The number is half the length of one rectangle side. Default = 0.5.

`capitalcoords` logical. If TRUE, checks if the coordinate fall within a rectangle around the capital of the country specified in x$country. The size of the rectangle can be controlled using the "countthresh" argument. Default = FALSE.

`capthresh` numeric. The size of the rectangle around the capital (in degrees). The number is half the length of one rectangle side. Default = 0.5.

`countrycheck` logical. If TRUE, checks if the coordinates fall within the country borders of the country indicated in x$country. Default = FALSE.

`polygons` The reference polygons for the countrycheck function. By default the wrld_simpl dataset from the `maptools` package. The `maptools` package must be loaded to use countrycheck = T.

`referencecountries` The reference coordinates for the country centroids and capitals. By default from the `countryref` data.

`outp` character defining the output values. See value section.

Details

The capital and country centroids in the country ref dataset are from the CIA World Factbook. The check for country borders is based on the world_simpl data from the `maptools` package. Please note that the ISO2 code for Namibia (“NA”) might cause problems with the countrycheck argument. If possible use ISO3 country codes.

Value

if outp = 'summary', a vector of the same length as the input data.frame with TRUE = clean coordinates, FALSE = suspicious coordinates. If outp = 'detailed', a data.frame with one column for each check that was performed: TRUE = clean coordinates, FALSE = suspicious coordinates. If outp = 'cleaned', a cleaned version of the input data.

Note

See the speciesgeocodeR documentation for further information and examples.

References

CENTRAL INTELLIGENCE AGENCY (2014) *The World Factbook*, Washington, DC.

http://opengeocode.org/download/cow.php

Examples

```
data(lemurs_test)
require(maptools)

#run all tests
data(wrld_simpl)
data(countryref)
test <- GeoClean(lemurs_test, GBIFhead = TRUE,
                 countrycentroid = TRUE, contthresh = 0.5,
 capitalcoords = TRUE, capthresh = 0.5,
 countrycheck = FALSE, outp = "cleaned")

insidecountry <- GeoClean(test, isna = FALSE, isnumeric = FALSE,
                          coordinatevalidity = FALSE,
  containszero = FALSE, zerozero = FALSE,
  latequallong = FALSE, GBIFhead = FALSE,
  countrycentroid = FALSE,
  contthresh = 0.5, capitalcoords = FALSE,
  capthresh = 0.5, countrycheck = TRUE,
  polygons = wrld_simpl)
#outp = "detailed"
test <- GeoClean(lemurs_test, GBIFhead = TRUE,
                 countrycentroid = TRUE, contthresh = 0.5,
 capitalcoords = TRUE, capthresh = 0.5,
 countrycheck = FALSE, outp = "detailed")
```

GetElevation *Elevation Data for Multiple Species*

Description

A wrapper using the getData function of the raster package to get elevation information for each occurrence point in x.

Usage

```
GetElevation(x)
```

Arguments

`x` an object of the class data.frame, spgeoIN, spgeoOUT or a character string.

Details

If `x` is a `data.frame`, it must have 3 columns: species name, longitude and latitude. If `x` is a character string it can be the path to a tab delimited .txt file with three columns: species name, longitude and latitude. If `x` is a character string of species names, the function will use rgbif to download occurrence information from GBIF. In this case no data cleaning is performed, except from the 'removeZeros' option of rgbif.

Value

A vector of elevation values.

See Also

`ReadPoints`, `SpGeoCod`.

Examples

```
data(lemurs)
## Not run:
GetElevation(lemurs)
## End(Not run)
```

`lemurs` *Distribution of Lemur Species*

Description

A `data.frame` with point occurrences of 39 Lemur species from Madagascar as example for the input format for the `GeoClean` function. The data is from GBIF.

Usage

```
data("lemurs")
```

Format

A data frame with 403 observations on the following 3 variables.

`identifier` species name

`XCOOR` longitude

`YCOOR` latitude

Source

`www.gbif.org`.

Examples

```
data(lemurs)
str(lemurs)
```

`lemurs_in` *Example for an spgeoIN Object*

Description

This is an example dataset for an object of the class spgeoIN (created by `ReadPoints`). It contains occurrence data for 39 species of Lemurs in Madagascar. The data has 3 slots: identifier = a `vector` of character strings containing the species name for each occurrence points species_coordinates = a `data.frame` with the longitude and latitude positions polygons = a `SpatialPolygons` object with a set of 3 polygons.

Usage

```
data("lemurs_in")
```

See Also

`ReadPoints`

Examples

```
data(lemurs_in)
summary(lemurs_in)
plot(lemurs_in)
```

`lemurs_test` *Distribution of Lemur Species with Problematic Records*

Description

A `data.frame` with point occurrences of 42 Lemur species in the right input format for the `GeoClean`. The data is downloaded from GBIF but has some added erroneous coordinates.

Usage

```
data("lemurs_test")
```

Format

A data frame with 627 observations on the following 4 variables.

`identifier` species name

`XCOOR` longitude

`YCOOR` latitude

`country` a factor with levels `MDG`

Source

`www.gbif.org`

Examples

```
data(lemurs_test)
str(lemurs_test)
```

MapGrid *Plotting Rasters in the Geographical Context*

Description

A wrapper function to map a raster with country borders. Can be used with the results of RichnessGrid.

Usage

```
MapGrid(rast, ...)
```

Arguments

rast	A raster object.
...	Additional arguments past to plot.

Note

See the speciesgeocodeR tutorial for further information and examples.

See Also

RichnessGrid

Examples

```
data(lemurs)
data(mdg_poly)

inp <- ReadPoints(lemurs, mdg_poly)
outp <- SpGeoCodH(inp)
e <- c(42, 52, -27, -10)
ras <- RichnessGrid(outp, e, reso = 60,  "abu")
MapGrid(ras)
```

MapRichness *Plotting Species Richness in a Set of Polygons*

Description

Illustrates the species number per polygon, based on and object of the class spgeoOUT (created by SpGeoCod).

Usage

```
MapRichness(x, areanames = NA, leg = "continuous",
            show.occ = F, lin.col = "grey", lwd = 1, ...)
```

Arguments

x	an object of the class ”spgeoOUT”.
areanames	a character string. If the polygons in the spgeoIN object were derived from a shape file, this column indicates the name of the column used for area names. If the polygons in the spgeoIN object were derived from a text file leave as default (Default = "").
leg	Controls the plot legend and coloring scheme. Must be either “continuous” or “discrete”, (default = continuous).
show.occ	logical. Defines if the occurrence points should be added to the plot (default = F).
lin.col	character string. Defines the colour of polygon borders (default = “grey”).
lwd	numeric. Line width of polygon lines.
...	Arguments to be passed to other methods, such as graphical parameters (see par).

Value

A plotted map.

Note

See the speciesgeocodeR tutorial for further information and examples.

See Also

RichnessGrid, MapGrid, RangeRichness.

Examples

```
data(lemurs)
data(mdg_poly)

inp <- ReadPoints(lemurs, mdg_poly)
outp <- SpGeoCodH(inp)
MapRichness(outp)
```

mdg_biomes *WWF Biomes of Madagascar*

Description

A SpatialPolygonsDataFrame of a simplified version of the WWF biomes of Madagascar.

Usage

```
data("mdg_biomes")
```

References

Olson, D. M., Dinerstein, E., Wikramanayake, E. D., Burgess, N. D., Powel, G. V. N., Underwood, E. C., Damico, J. A.,Itoua, I., Strand, H. E., Morrison, J. C., Loucks, C. J., Ricketts, T. H., Kura, Y., Lamoreux, J. F., Wettengel, W. W., Hedao, P., and Kassem, K.R. 2001 Terrestrial ecoregions of the world: A new map of life on earth. *BioScience*, **51**(11):933–938.

http://www.worldwildlife.org/publications/terrestrial-ecoregions-of-the-world

Examples

```
data(mdg_biomes)
str(mdg_biomes)
```

mdg_poly *WWF Biomes of Madagascar in Table Format*

Description

A data.frame of a simplified version of the WWF biomes of Madagascar (simplified).

Usage

```
data("mdg_poly")
```

Format

A data frame with 87 observations on the following 3 variables.

identifier a factor with levels Dry Forest Moist Forest Shrublands

XCOOR a numeric vector

YCOOR a numeric vector

References

Olson, D. M., Dinerstein, E., Wikramanayake, E. D., Burgess, N. D., Powel, G. V. N., Underwood, E. C., Damico, J. A.,Itoua, I., Strand, H. E., Morrison, J. C., Loucks, C. J., Ricketts, T. H., Kura, Y., Lamoreux, J. F., Wettengel, W. W., Hedao, P., and Kassem, K.R. 2001 Terrestrial ecoregions of the world: A new map of life on earth. *BioScience*, **51**(11):933–938.

http://www.worldwildlife.org/publications/terrestrial-ecoregions-of-the-world

Examples

```
data(mdg_poly)
str(mdg_poly)
```

plot.spgeoIN *Plot Method for SpgeoIN Objects*

Description

plot.spgeoIN is the plotting method for objects of the class spgeoIN. It is designed to give an overview of the data and plots the input points and polygons on a world map.

Usage

```
## S3 method for class 'spgeoIN'
plot(x, ...)
```

Arguments

`x`	an object of the class spgeoIN
`...`	arguments to be passed to methods, such as graphical parameters (see par).

See Also

ReadPoints

Examples

```
data(lemurs)
data(mdg_poly)

inp <- ReadPoints(lemurs, mdg_poly)
plot(inp)
```

plot.spgeoOUT *Plot Method for SpgeoOUT Objects*

Description

plot.spgeoOUT is the plotting method for objects of the class spgeoOUT. Depending on the plottype argument the function can be used to get an overview over the data or plot all elements of the object.

Usage

```
## S3 method for class 'spgeoOUT'
plot(x,  plottype = "summary", plotout = F,
                        mode = c("percent", "total"),
moreborders = F, areanames = NULL, ...)
```

Arguments

`x` an object of the class spgeoOUT

`plottype` a character string. Controls which data shall be plotted. Default is to “summary”. See details for more information.

`plotout` logical. If TRUE, no user-input is needed between the plots (par(ask = FALSE)), if FALSE, user input is needed to browse between plots (par(ask = TRUE)).

`mode` a character string. Can be “percent” or “total” controls if percentage or total is shown for `plottype` = “species”. Default is “percent”.

`moreborders` logical. If TRUE additional borders are added to the maps. Default is FALSE.

`areanames` character. character string. If the polygons of the spgeoOUT object were derived from a shape file, this column indicates the name of the column used for area names. See the SpeciesgeocodeR documentation for more information, if the polygons of the spgeoOUT object were derived from a text file leave as default (Default = "").

`...` arguments to be passed to methods, such as graphical parameters (see `par`).

Details

The output will be different depending on the plottype argument. `plottype` = “summary”: a map with all occurrence points and polygons (the red points were not classified to any polygon), the species number for all polygons, and if available, a heatplot visualizing the coexistence matrix; `plottype` = “species”: a bar chart for each input species showing the number of occurrences in each polygon; `plottype` = “polygons”: a bar chart for each input-polygon showing the number of occurrences for each species in the polygon; `plottype` = “speciesrichness”: a bar chart showing species number per polygon; `plottype` = “coexistence”: a heatplot visualizing the coexistence matrix. The heatcolors code for the percent of occurrences that are shared (per rows); `plottype` = “mapspecies”: one map per species showing all occurrence points in all polygons; `plottype` = “mappolygons”: a series of maps, showing each polygon and its close environment, with all samples classified to this polygon. Species are color-coded; `plottype` = “mapunclassified”: a map with all sample points that could not be classified to any polygon; `plottype` = “mapall”: a map of all occurrence points in the object.

See Also

`summary.spgeoOUT`, `SpGeoCodH`

Examples

```
data(lemurs)
data(mdg_poly)

inp <- ReadPoints(lemurs, mdg_poly)
outp <- SpGeoCodH(inp)
plot(outp)
plot(outp, plottype = "species")
plot(outp, plottype = "polygons")
plot(outp, plottype = "speciesrichness")
plot(outp, plottype = "coexistence")
plot(outp, plottype = "mapspecies")
plot(outp, plottype = "mappolygons")
plot(outp, plottype = "mapunclassified")
plot(outp, plottype = "mapall")
```

PlotHull *Plotting Species Ranges*

Description

Plots species distribution ranges derived with CalcRange in the geographic context.

Usage

```
PlotHull(dat, xlim = c(-180, 180), ylim = c(-90, 90),
         col = rgb(255, 0, 0, 50, maxColorValue = 255),
 border = rgb(255, 0, 0, 50, maxColorValue = 255),
 type = "overlay", select = "all", ...)
```

Arguments

dat	a list of ‘SpatialPolygons’ objects created with CalcRange.
xlim	vector indicating the x-axis plotting limits. Default is c(-180, 180).
ylim	a vector indicating the y-axis plotting limits. Default = c(-90, 90).
col	character string indicating the colour of the polygons. Default is to transparent red.
border	a character string indicating the color of the polygon borders. Default is to transparent red.
type	a character string indicating if all range polygons in the input file shall be overlaid or plotted on separate plots. Must be “seperate” or “all”. Default is to “all” (overlay).
select	a character string indicating which elements of the list to plot. Can be one or multiple species names. Default = “all”.
...	additional arguments passed to plot.

Details

If `type` is set to “separate”, set `par(ask = T)` to see all maps.

Value

A plot or a set of plots.

Examples

```
data("lemurs_in")
dat <- CalcRange(data.frame(lemurs_in$identifier,
                             lemurs_in$species_coordinates),
                 value = "shape")
PlotHull(dat, xlim = c(-130, -100), ylim = c(30,60))
```

RangeRichness *Species Richness Raster from Range Polygons*

Description

Calculate species richness for the specified grid based on an list of range polygons, as derived from `CalcRange`.

Usage

```
RangeRichness(ra, limits = c(-180, 180, -90, 90), reso = 60)
```

Arguments

`ra`	A list of SpatialPolygons with species ranges.
`limits`	numeric. A vector with the limits of the grid in decimal degrees of the format c(xmin, xmax, ymin, ymax). Default is worldwide c(-180, 180, -90, 90).
`reso`	numeric. resolution of the grid in minutes.

Value

a `raster` object.

Examples

```
data(lemurs)
rang <- CalcRange(lemurs, value = "shape")
sprich <- RangeRichness(rang)
MapGrid(sprich)
```

ReadPoints *Loading Data into SpeciesgeocodeR*

Description

Create an object of the class spgeoIN from the input arguments.

Usage

```
ReadPoints(x, y, areanames = NA, verbose = FALSE, cleaning = FALSE)
```

Arguments

`x`	a `data.frame` with three columns named identifier, XCOOR, YCOOR or a `character string`. If x is a `character string`, it can either be the path to a tab-delimited text file containing the coordinates of points of interest (tab delimited, three columns with header: identifier, XCOOR, YCOOR) OR a vector of species names. In the latter cases GBIF is searched for occurrence data using the rgbif function of the rgbif package.

`y` character string giving the path to a text file containing the polygons of points of interest (tab delimited, three columns with header: identifier, XCOOR, YCOOR) OR an object of the class `SpatialPolygons` OR an object of the class `SpatialPolygonsDataFrame`.

`areanames` character string. If the polygons in the spgeoIN object were derived from a shape file or a SpatialPolygonsDataframe, this column indicates the name of the column used for area names. See the Speciesgeocoder tutorial for more information. If the polygons in the spgeoIN object were derived from a text file leave as default (Default = NA).

`verbose` logical. If TRUE, the functions report the progress of the analyses. This is advisable for large datasets.

`cleaning` logical. If TRUE, `GeoClean` is applied to the input coordinates. Default = FALSE.

`...` Further arguments to be passed to `GeoClean`.

Value

An object of class spgeoIN.

Note

See the speciesgeocodeR tutorial for further information and examples.

See Also

`lemurs_in`

Examples

```
data(lemurs)
data(mdg_poly)

inp <- ReadPoints(lemurs, mdg_poly)
```

`RichnessGrid` *Species Richness and Abundance Grids from Point Records*

Description

Create a species richness or abundance grid, from a spgeoOUT object or a data.frame with species names and occurrence coordinates.

Usage

```
RichnessGrid(x, limits = c(-180, 180, -90, 90), reso,
             type = c("spnum", "abu"))
```

Arguments

x — an object of the class `data.frame` or `spgeoIN` or `spgeoOUT` or `character string`. If x is a `data.frame`, it must consist of three columns, indicating species name, longitude and latitude. If x is a character string, it can be the name of a tab delimited text file, with three columns indicating species name, longitude and latitude or a vector of species names. In the latter case the GBIF database is search for geo-referenced occurrences.

limits — numerical. A vector of the form (xmin, xmax, ymin, ymax) givin the raster limits in decimal degrees.

reso — numeric. resolution of the grid in minutes.

type — character string indicating the type of data displayed: “spnum” = number of species per grid cell, “abu” = the number of occurrence points per grid cell.

Value

a `raster` object.

See Also

`MapRichness`, `MapGrid`, `RangeRichness`.

Examples

```
# for x = data.frame
data(lemurs)
e <- c(-125, -105, 30, 50)
dat <- RichnessGrid(lemurs, e, reso = 60, type = "spnum")

# for x = character string
e <- c(-125, -105, 30, 50)

## Not run:
dat <- RichnessGrid("Indri indri", e, reso = 60, type = "spnum")

## End(Not run)
```

SpeciesGeoCoder — *A Complete SpeciesgeocodeR Analysis*

Description

A complete occurrence to area classification with a set of standard output files saved to the working directory.

Usage

```
SpeciesGeoCoder(x, y, coex = FALSE, graphs = TRUE, areanames = "",
                  occ.thresh = 0, elevation = FALSE, threshold,
verbose = FALSE, cleaning = FALSE, ...)
```

Arguments

`x` a `data.frame` with three columns named identifier, XCOOR, YCOOR or a `character string`. If x is a character string it can be the path to a tab-delimited text file containing the coordinates of points of interest (tab delimited, three columns with header: identifier, XCOOR, YCOOR) OR a shape file OR a vector of species names. In the latter cases GBIF is search for occurrence data using the `rgbif` function.

`y` a `character string` giving the path to a text file containing the polygons of points of interest (tab delimited, three columns with header: identifier, XCOOR, YCOOR) OR an object of the class SpatialPolygons OR an object of the class `SpatialPolygonsDataframe`.

`coex` logical. Indicating if a coexistence matrix should be calculated and shown as a heat plot (default = FALSE). Only for small to medium-sized datasets.

`graphs` logical. Defines if the summary plots and maps are created as pdf files in the working directory (default = TRUE).

`areanames` a `character string`. If the input polygon file is a shape and not a text file, this column indicates the name of the column used for area names. See the SpeciesgeocodeR documentation for more information, if the polygon input file is a text file leave as default (default = "").

`occ.thresh` numerical. Threshold for the minimum number of occurrences necessary in a polygon to be counted as present (in percent of the total occurrences of the species). Default = 0.

`elevation` logical. Should the results be split by elevation class? (default = FALSE)

`threshold` numeric vector. Indicating the elevation thresholds.

`verbose` logical. If TRUE, reports the progress of the analyses.

`cleaning` logical. If TRUE, `GeoClean` is applied to the input coordinates.

`...` Further arguments to be passed to `GeoClean`.

Details

This performs a complete speciesgeocodeR occurrences to area classification analysis and produces the standard set of output files: 1. classification of all samples to a polygon, 2. Summary of species (identifier) occurrence per polygon, 3. a table of samples that could not be classified to any of the input polygons, 4. a nexus-file, including the species classification, 5. a coexistence matrix, showing to which percentage species to co-occur in the input polygons, 5. a table giving species numbers per polygon. Furthermore produces a set of .pdf files in the output directory: 1. a bar chart showing the number of species per polygon, 2. a bar chart summarizing numbers of each species for each polygon, 3. a bar chart summarizing the occurrences in each polygon per species, 4. a map of all polygons with the points classified to them, colored for species (identifier), 5. a map of the occurrences of all species, a map showing all points used and all unclassified points in the geographic context, 6. a heat plot showing the co-occurrence patterns of all species in the polygons. If elevation = T, the input dataset is split at the elevation(s) provided by threshold, and separate output files are generated for each elevation class.

Value

A set of output tables (.txt) and graphics (.pdf).

Note

See the speciesgeocodeR documentation for further information and examples.

Examples

```
## Not run:
data(lemurs)
data(mdg_poly)
SpeciesGeoCoder(lemurs, mdg_poly
## End(Not run)
```

`Spgc2BioGeoBEARS` *Geography Input for BioGeoBEARS*

Description

Convert an object of the class `spgeoOUT` in the right format to be used as geography input for BioGeoBEARS.

Usage

```
Spgc2BioGeoBEARS(x, phyl = NULL, file = NULL, true.areas = T, true.species = T)
```

Arguments

`x` an object of the clas `'spgeoOUT'`

`phyl` an object of the class `phylo`. An optional phylogeny to be used with BioGeoBEARS, to align species.

`file` character. The complete path and filename to the outputfile.

`true.areas` logical. If TRUE, areas with 0 species are removed from the matrix. Default = TRUE.

`true.species` logical. If TRUE, species not occurring in any area are removed from the matrix. Default = TRUE.

Details

This will create a file in the working directory.

Value

A file in the working directory and a list.

Author(s)

Alexander Zizka, Ruud Scharn

Examples

```
## Not run:
data(lemurs)
data(mdg_poly)

outp <- SpGeoCod(lemurs, mdg_poly)
conv <- Spgc2BioGeoBEARS(outp)

## End(Not run)
```

SpGeoCod   *Point to Polygon Classification from Text Files*

Description

Wrapper around ReadPoints and SpGeoCodH, runs a complete speciesgeocodeR occurrences to area classification analysis from text files as input.

Usage

```
SpGeoCod(x, y, areanames, occ.thresh = 0, elevation, threshold, cleaning = FALSE,
          ...)
```

Arguments

x — a character string giving the path to a text file containing the coordinates of points of interest (tab delimited, three columns with header: identifier, XCOOR, YCOOR) OR a data.frame with three columns named: identifier, XCOOR, YCOOR.

y — a character string giving the path to a shape file containing the polygons of interet OR a text file containing the polygons of interest (tab delimited, three columns with header: identifier, XCOOR, YCOOR) OR an object of the class SpatialPolygons OR an object of the class SpatialPolygonsDataframe.

areanames — a character string. If the input polygon file is a shape and not a text file, this column indicates the name of the column used for area names. See the SpeciesgeocodeR tutorial for more information, if the polygon input file is a text file leave as default (Default = "").

occ.thresh — numerical. Threshold for the minimum number of occurrences necessary in a polygon to be counted as present (in percent of the total occurrences of the species). Default = 0.

elevation — logical. Should the results be split by elevation class? (default = FALSE)

threshold — numeric vector. Indicating the elevation thresholds.

cleaning — logical. If TRUE, GeoClean is applied to the input coordinates.

... — Further arguments to be passed to GeoClean.

Value

An object of the class spgeoOUT.

Note

See the speciesgeocodeR tutorial for further information and examples.

See Also

ReadPoints, SpGeoCodH.

Examples

```
data(lemurs)
data(mdg_poly)

outp <- SpGeoCod(lemurs, mdg_poly)
```

SpGeoCodH *A Standard a SpeciesgeocodeR Area Classification*

Description

Run a standard speciesgeocodeR occurrence point to area classification analysis on an object of the class spgeoIN (created by ReadPoints). The results are stored in an object of the class spgeoOUT. See details for more information.

Usage

```
SpGeoCodH(x, areanames = NULL, occ.thresh = 0)
```

Arguments

x	an object of the class spgeoIN
areanames	a character string. If the polygons in the spgeoIN object were derived from a shape file, this column indicates the name of the column used for area names. See the Speciesgeocoder tutorial for more information. If the polygons in the spgeoIN object were derived from a text file leave as default (Default = NA).
occ.thresh	numerical. The threshold for the minimum number of occurrences necessary in an area to be counted as present (in percent of the total occurrences of the species). Default = 0.

Details

This function uses an object of the class spgeoIN and performs a point in polygon test classifying each species to a polygon, summarizes the information per samples that could not be classified and calculates a coexistence matrix. These objects are then put together with the input information to an object of the class spgeoOUT.

Value

An object of the class spgeoOUT.

Note

See the speciesgeocodeR tutorial for further information and examples.

Examples

```
data(lemurs)
data(mdg_poly)

inp <- ReadPoints(lemurs, mdg_poly)
outp <- SpGeoCodH(inp)
names(outp)
```

summary.spgeoIN *Summary Method for SpgeoIN*

Description

Summarizes the content of a spgeoIN object.

Usage

```
## S3 method for class 'spgeoIN'
summary(object, areanames = NA,  ...)
```

Arguments

`object`	object of class spgeoIN
`areanames`	character. Indicating the column with the area names, if not specified, the ID slot is displayed.
`...`	arguments to be passed to methods, such as graphical parameters (see par).

Details

The summary gives an overview of the number of species, occurrence points and number of polygons in the object, and gives a summary of the input coordinates as well as the polygon names.

See Also

`ReadPoints`

Examples

```
data(lemurs)
data(mdg_poly)
inp <- ReadPoints(lemurs, mdg_poly)
summary(inp)
```

summary.spgeoOUT *Summary Method for SpgeoOUT*

Description

summary method for objects of the class spgeoOUT

Usage

```
## S3 method for class 'spgeoOUT'
summary(object, areanames = NA, ...)
```

Arguments

object	an object of the class spgeoOUT
areanames	a character string. Indicating the column containing the area names, if not specified the ID slot is displayed.
...	arguments to be passed to methods, such as graphical parameters (see par).

Details

The summary gives an overview of the number of species, occurrence points and number of polygons in the object, and gives a summary of the input coordinates as well as the polygon names. Additionally it shows a summary of species diversity per polygon and the number of species not classified to any polygon.

See Also

SpGeoCodH

Examples

```
data(lemurs)
data(mdg_poly)

inp <- ReadPoints(lemurs, mdg_poly)
outp <- SpGeoCodH(inp)
summary(outp)
```

WriteOut *Write SpeciesgeocodeR results to the Working Directory*

Description

This function writes the results of a speciesgeocodeR analysis to the working directory. Output statistics as .txt files, maps and plots as .pdf and a nexus file for use in phylogenetic analyses.

Usage

```
WriteOut(x,writetype = c("all", "BioGeoBEARS", "coexistence", "graphs",
                          "maps", "nexus","statistics"), areanames = NULL)
```

Arguments

`x` an object of the class spgeoOUT

`writetype` a character string. Defines which results should be written to disk. Default is to “all”, see details for other options

`areanames` a character string. If the polygons of the spgeoOUT object were derived from a shape file, this column indicates the name of the column used for area names. See the SpeciesgeocodeR documentation for more information, if the polygons of the spgeoOUT object were derived from a text file leave as default.

Details

There are different options for the `writetype` argument: “all” = all summary tables, plots, maps and a heatplot of species-coexistence if calculated (see `CoExClass`); “BioGeoBEARS” = a text file in the geography input format of BioGeoBEARS; “coexistence” = the coexistence matrix as table and heatplot “graphs” = summary graphs as pdf files; “maps” = summary maps as .pdf file; “nexus” = write out the species to area classification as nexus file; “statistics” = the summary tables as .txt files.

Value

a set of files in the working directory

See Also

`SpeciesGeoCoder`, `SpGeoCod`, `SpGeoCodH`

Examples

```
## Not run:
data(lemurs)
data(mdg_poly)

inp <- ReadPoints(lemurs, mdg_poly)
outp <- SpGeoCodH(inp)
WriteOut(outp)
WriteOut(outp, writetype = "graphs")
WriteOut(outp, writetype = "statistics")
WriteOut(outp, writetype = "nexus")
WriteOut(outp, writetype = "coexistence")

## End(Not run)
```

`WwfLoad` *Load the WWF Terrestrial Ecoregions*

Description

Downloads a shape-file containing the terrestrial ecoregions as defined by Olson et al. 2001 from worldwildlife.org/publications/terrestrial-ecoregions-of-theworld, unpacks them and loads the into R.

Usage

```
WwfLoad(x = "")
```

Arguments

x the path to the folder where the shapes should be saved. If set to "" the files are saved to the working directory.

Value

An object of the class `SpatialPolygonsDataframe`

Note

See the speciesgeocodeR documentation for further information and examples.

References

Olson, D. M., Dinerstein, E. ,Wikramanayake, E. D., Burgess, N. D., Powel, G. V. N., Underwood, E. C., Damico, J. A.,Itoua, I., Strand, H. E., Morrison, J. C., Loucks, C. J., Ricketts, T. H., Kura, Y., Lamoreux, J. F., Wettengel, W. W., Hedao, P., and Kassem, K.R. 2001 Terrestrial ecoregions of the world: A new map of life on earth. *BioScience*, **51**(11):933–938.

http://www.worldwildlife.org/publications/terrestrial-ecoregions-of-the-world

Examples

```
## Not run:
wwf_eco <- WwfLoad()
plot(wwf_eco)
names(wwf_eco)
## End(Not run)
```

Index