

Package ‘ENiRG’

October 13, 2016

Type Package

Title Ecological Niche in R and GRASS

Version 1.0.1

Date 2016-10-12

Author Fernando Canovas, Chiara Magliozzi, Jose Antonio Palazon-Ferrando, Frederico Mestre, Mercedes Gonzalez-Wanguemert

Maintainer Fernando Garcia Canovas <fcgarcia@ualg.pt>

Description A set of tools for the analysis of ecological niche of species and calculation of habitat suitability maps.

License GPL (>= 2)

Depends ade4, miniGUI, raster, sp, rgrass7

Imports gdata, fgui, tcltk

NeedsCompilation no

Repository CRAN

Date/Publication 2016-10-13 00:33:26

R topics documented:

ENiRG-package	2
apis.enirg	4
apis.hsm	5
boyce	6
cbi.results	8
classify.map	9
enirg	10
enirg.GUI	14
enirg.plot	14
enirg.predict	16
import.egvs	18
list.maps	19
map.info	21
stdz.maps	22

Description

ENiRG has been designed to characterize the realized niche of the species by interfacing R software with GRASS geographical information system in order to overcome issues when working with large data sets (i.e., wide areas or high resolution). This package uses classes defined in `rgrass7` package to deal with spatial data and to interface R and Grass.

Details

Package: ENiRG
 Type: Package
 Version: 1.0-1
 Date: 2016-05-03
 License: GPL (>=2)

This package is organised in four main parts:

- data preparation (`import.egvs`, `list.maps`, `map.info` and `stdz.maps`);
- ENFA analysis (`enirg` and `enirg.plot`);
- prediction of species' niche and HSM calculation (`enirg.predict`);
- classification of suitability (`boyce` and `classify.map`).

ENiRG is able to perform the analysis into GRASS, by initiating a session within R (see also function `initGRASS` from `rgrass7`). *ENiRG* provides an easy way to import raster maps through `import.egvs`, containing EGV's distribution in any of the formats supported by GDAL library (`gdal.org`). It also provides several functions to explore GRASS environment, giving the available list of maps (`list.maps`) and metadata (`map.info`). Before run ENFA analysis, standardization of quantitative EGVs is suggested Hirzel *et al.* (2002). `stdz.maps` allows the process by retrieving its univariate statistics.

Characterization of the species's niche can be done by using the main function of the package. `enirg` performs ENFA analysis, while projection of the ordination diagram of marginality and specialization (ENFA's principal components), can be computed using `enirg.plot`.

Function `enirg.predict` focuses on computing the prediction of the species' niche accordingly with ENFA's results.

Main functions of the package offer two methods, accordingly to the amount of data computed. User can choose one of two available methods: "normal" and "large". The first, strongly relies on the `rgrass7` package and thus is limited by R storage capacity and computations; on the contrary, "large" method directly interface with GRASS, allowing calculations over large areas or high resolution maps with huge amount of data (NOTE: it is only available for Linux/Unix OS, at the moment).

`boyce` function aims to investigate how accurately the map obtained from function `enirg.predict` is predicting modelled species presences (Boyce *et al.*, 2002). Intervals estimated manually, allow users to reclassify predicted niche maps, by using `classify.map`, and thus distinguishing unsuitable, marginal, suitable and optimal habitat (Hirzel *et al.*, 2006). Evaluation of habitat suitability model accuracy is made by means of n-fold cross-validation (Fielding and Bell, 1997).

User can take advantages of integration with other R libraries (*raster*, *rasterVIS*), portability and interoperability within GRASS (i.e. efficient map storage) and can also communicate with other commonly used GIS software, such as QGIS.

A graphical user interface (GUI) allows better access to functionalities of *ENiRG* package through function `link{enirg.GUI}`.

Note

The package depends on `rgrass7`, `raster`, `R.utils`, `stats`, `gplots`, `miniGUI`, `ade4`, `tcltk2`, `fgui`.

Author(s)

Fernando Canovas, Chiara Magliozzi, Jose Antonio Palazon-Ferrando, Frederico Mestre, Mercedes Gonzalez-Wanguemert

Maintainer: Fernando Garcia Canovas <fcgarcia@ualg.pt>

References

Boyce, M.S., Vernier, P.R., Nielsen, S.E., Schmiegelow, F.K.A. (2002). Evaluating resource selection functions. *Ecological Modelling* 157, 281-300.

Fielding, A., Bell, J. (1997). A review of methods for the assessment of prediction errors in conservation presence/absence models. *Environmental Conservation*, 24, 38-49.

Hirzel, A.H., Hausser, J., Chessel, D. and Perrin, N. (2002). Ecological-niche factor analysis: How to compute habitat-suitability maps without absence data? *Ecology*, 83, 2027-2036.

Hirzel, A.H., Le Lay, G., Helfer, V., Randin, C., Guisan, A. (2006). Evaluating the ability of the habitat suitability models to predict species presences. *Ecological Modelling* 199, 142-152.

Canovas, F., Magliozzi, C., Mestre, F., Palazon-Ferrando, J.A. and Gonzalez-Wanguemert, M. (2016). ENiRG: R-GRASS interface for efficiently characterizing the ecological niche of species and predicting habitat suitability. *Ecography* 39: 593-598.

See Also

`adehabitatHS`

apis.enirg

List of an object enirg.

Description

apis.enirg is a list which stores the output of an analysis performed with [enirg](#) over occurrence records of the *Apis mellifera* species across the Iberian Peninsula (Canovas et al., 2002, 2008, 2014).

Usage

```
data(apis.enirg)
```

Details

The source dataset only refers to the African lineage occurrence records.

References

Canovas, F., De la Rúa, P., Serrano, J. and Galian, J. (2002). Mitochondrial DNA variability in *Apis mellifera* iberica populations from Galicia. *Archivos de Zootecnia*, 51: 441-448

Canovas, F., De la Rúa, P., Serrano, J. and Galian, J. (2008). Geographical patterns of mitochondrial DNA variation in *Apis mellifera* iberiensis (Hymenoptera: Apidae). *Journal of Zoological Systematics and Evolutionary Research*, 46(1): 23-30

Canovas, F., De la Rúa, P., Serrano, J. and Galian, J. (2014). Analysing and describing a contact area of two distinct evolutionary units. An ecological perspective. *Journal of Insect Conservation*, 18(5): 927-937

Canovas, F., Magliozzi, C., Mestre, F., Palazon-Ferrando, J.A. and Gonzalez-Wanguemert, M. (2016). ENiRG: R-GRASS interface for efficiently characterizing the ecological niche of species and predicting habitat suitability. *Ecography* 39: 593-598.

Examples

```
require(raster)

# loading map of Iberian Peninsula
mask.file <- paste(system.file(package = "ENiRG"),
                  "/ext/", "mask.asc", sep="")
plot(raster(mask.file), col="brown", legend=FALSE, xlab = "longitude",
     ylab = "latitude")

data(apis.enirg)
points(apis.enirg$presences[, 1:2],
       cex=apis.enirg$presences[, 3] / 30, pch=19)
```

apis.hsm *List with values returned from [enirg.predict](#).*

Description

apis.hsm is a list which stores output of prediction for the African lineage of honeybees in the Iberian Peninsula (Canovas *et al.*, 2002, 2008, 2014).

Usage

```
data(apis.hsm)
```

Details

A list of:

- predictions. A data frame with observed and predicted values.
- validation. A data frame with the distribution of predicted values for both the observed data and the entire predicted map.
- map. Prediction map for the African lineage of honeybees in the Iberian Peninsula.

References

Canovas, F., De la Rúa, P., Serrano, J. and Galian, J. (2002). Mitochondrial DNA variability in *Apis mellifera iberica* populations from Galicia. *Archivos de Zootecnia*, 51: 441-448

Canovas, F., De la Rúa, P., Serrano, J. and Galian, J. (2008). Geographical patterns of mitochondrial DNA variation in *Apis mellifera iberiensis* (Hymenoptera: Apidae). *Journal of Zoological Systematics and Evolutionary Research*, 46(1): 23-30

Canovas, F., De la Rúa, P., Serrano, J. and Galian, J. (2014). Analysing and describing a contact area of two distinct evolutionary units. An ecological perspective. *Journal of Insect Conservation*, 18(5): 927-937

Examples

```
data(apis.hsm)

require(raster)

plot(apis.hsm$African_predicted_hsm)
contour(apis.hsm$African_predicted_hsm, add = TRUE)
```

 boyce

Habitat suitability classification using continuous Boyce index (CBI).

Description

boyce.classification function aims to investigate accuracy of the map obtained from function `enirg.predict`, by using the continuous Boyce index (CBI; Boyce *et al.*, 2002).

Curves from P/E ratio give the possibility to reclassify the predicted niche map (`enirg.predict`) and thus distinguish unsuitable, marginal, suitable and optimal habitats. In accordance with the method proposed by Hirzel *et al.*(2006), it interprets predicted-to-expected ratio (P/E) by partitioning habitat suitability predictions into classes and by calculating frequencies. If model properly delineates suitable areas for the studied species, Spearman rank correlation coefficient of the ratio F_i , will be 1.

Evaluation of habitat suitability model accuracy is made by means of n-fold cross-validation (Fielding and Bell, 1997), partitioning data evenly but randomly into `cv.sets` partitions. Once ratio is calculated, Spearman correlation coefficient allows to estimate fitting for the predicted-to-expected relationship.

Categories allows to use function `classify.map`, in order to perform a classification on the HSM (`enirg.predict`).

Usage

```
boyce(prediction, prediction.map, categories = NULL, cv.sets = 10,
      type = "manual", outcat = "cbi.results")
```

Arguments

<code>prediction</code>	vector. Predicted suitability values from observations or from a second validation data set.
<code>prediction.map</code>	vector. Predicted suitability values for the entire area of study.
<code>categories</code>	vector with desired categories. NULL if <code>type = "manual"</code>
<code>cv.sets</code>	integer, indicating the number of subsets to use for the cross validation.
<code>type</code>	string. If "manual", a GUI assists the classification process. If "none", <code>categories</code> parameter will be used.
<code>outcat</code>	string. Name for object which will contain the results.

Details

A GUI allows a manual adjustment of suitability classes. An ideal model would give a straight P/E curve. Curve shape and its confidence interval can be used to define boundaries of habitat suitability classes (as suggested by vertical dashed lines).

Value

This function displays predicted/expected ratio curve shapes. Also `enirg.predict` returns a list object, containing the following components:

- coefficients. A vector of two: spearman rank coefficient and adjusted r squared.
- intervals. Suitability intervals for later using with [classify.map](#)

Author(s)

Fernando Canovas <fcgarcia@ualg.pt>

References

Boyce, M.S., Vernier, P.R., Nielsen, S.E., Schmiegelow, F.K.A. (2002). Evaluating resource selection functions. *Ecological Modelling* 157, 281-300.

Fielding, A., Bell, J. (1997). A review of methods for the assessment of prediction errors in conservation presence/absence models. *Environmental Conservation*, 24, 38-49.

Hirzel, A.H., Le Lay, G., Helfer, V., Randin, C., Guisan, A. (2006). Evaluating the ability of the habitat suitability models to predict species presences. *Ecological Modelling* 199, 142-152.

See Also

[enirg.predict](#), [predict.enfa](#)

Examples

```
data(apis.hsm)

# vector of predictions for observations:
apis.predictions <- apis.hsm$predictions[, 2]

# vector of predictions for the entire predicted map:
apis.predictions.map <- as.vector(na.exclude(apis.hsm$African_predicted_hsm@data@values))

# Try with intervals:
# unsuitable = 0.65
# marginal = 0.84
# suitable = 0.96
# Note that this species has an optimal distribution
# in the study area, resulting in a wide unsuitable
# interval and narrow suitable ones.

boyce(prediction = apis.predictions,
       prediction.map = apis.predictions.map,
       categories = c(0, 0.65, 0.84, 0.96, 1),
       cv.sets = 10, type = "none")
```

`cbi.results`*List with values returned from `boyce`.*

Description

`cbi.results` is a list which stores output of the continuous Boyce classification performed with `boyce`.

Usage

```
data(cbi.results)
```

Details

Source dataset refers to classification of the niche analysis performed on `apis.enirg` data.

References

Boyce, M.S., Vernier, P.R., Nielsen, S.E., Schmiegelow, F.K.A. (2002). Evaluating resource selection functions. *Ecological Modelling* 157, 281-300.

Fielding, A., Bell, J. (1997). A review of methods for the assessment of prediction errors in conservation presence/absence models. *Environmental Conservation*, 24, 38-49.

Hirzel, A.H., Le Lay, G., Helfer, V., Randin, C., Guisan, A. (2006). Evaluating the ability of the habitat suitability models to predict species presences. *Ecological Modelling* 199, 142-152.

Examples

```
data(apis.hsm)

# vector of predictions for observations:
apis.predictions <- apis.hsm$predictions[, 2]

# vector of predictions for the entire predicted map:
apis.predictions.map <- as.vector(na.exclude(apis.hsm$African_predicted_hsm@data@values))

# Try with intervals:
# unsuitable = 0.65
# marginal = 0.84
# suitable = 0.96
# Note that this species has an optimal distribution
# in the study area, resulting in a wide unsuitable
# interval and narrow suitable ones.

boyce(prediction = apis.predictions,
       prediction.map = apis.predictions.map,
       categories = c(0, 0.65, 0.84, 0.96, 1),
       cv.sets = 10, type = "none")
```



```
data(cbi.results)
cbi.results
```

`classify.map`*Classification of the HSM*

Description

`classify.map` classifies the Habitat Suitability Map (HSM) using suitability classes.

Usage

```
classify.map(map, suit.classes, output.name = NULL, load.map = FALSE)
```

Arguments

<code>map</code>	unclassified suitability map name from GRASS.
<code>suit.classes</code>	object of class boyce , giving the suitability classes.
<code>output.name</code>	string. Name for classified map.
<code>load.map</code>	logical. Whether the map should be uploaded as an object of class raster

Value

Function `classify.map` allows to classify the HSM according to breaks, which represents limits for intervals in the four suitability classes (unsuitable, marginal, suitable and optimal; Hirzel et al., 2006). Those values can be obtained by using CBI [boyce](#). HSM classified map layer in GRASS can be also uploaded into R by setting `load.map` to TRUE.

Author(s)

Fernando Canovas <fcgarcia@ualg.pt>

References

Hirzel, A.H., Le Lay, G., Helfer, V., Randin, C., Guisan, A. (2006). Evaluating the ability of the habitat suitability models to predict species presences. *Ecological Modelling* 199, 142-152.

See Also

[enirg](#), [enirg.predict](#)

Examples

```
## Not run:
# starting GRASS session
initGRASS("/usr/bin/grass-7.0.0", home=tempdir())
initGRASS("C:/GRASS", home=tempdir())

data(apis.enirg)

# presences table
lina <- apis.enirg$presences

# loading the environmental information in batch
predictor.names <- c("tann", "mxtwm", "mntcm", "rfdm", "rfseas")
predictor.maps <- paste("std_", predictor.names, sep="")
file.names <- paste(system.file(package = "ENiRG"),
                    "/ext/", predictor.names, ".asc", sep="")

import.egvs(file.names, predictor.names)

# standardization
stdz.maps(predictor.names, predictor.maps)

# performing the Ecological Niche Factor Analysis (ENFA)
enirg(presences.table = lina, qtegv.maps = predictor.maps,
      species.name = "African", nf = 3, scannf = FALSE,
      method = "normal") -> apis.enfa

enirg.predict(apis.enfa, load.map = TRUE, method = "normal") -> apis.hsm

data(cbi.results)

classify.map(map = "African_predicted_hsm",
             suit.classes = cbi.results,
             output.name = "African_hsm_class") -> apis.hsm.class

## End(Not run)
```

enirg

Ecological Niche in R-Grass

Description

enirg performs an Ecological Niche Factor Analysis (ENFA) by coupling R and GRASS softwares and following Hirzel *et al.* (2002).

Usage

```
enirg(presences.table, qtegv.maps, qllegv.maps = NULL, col.w = NULL,
      scannf = TRUE, nf = 1, method = "normal", load.maps = FALSE,
      species.name = "species", map.center = NULL, res = NULL)
```

Arguments

<code>qtegv.maps</code>	vector of strings, giving names of raster maps, containing quantitative environmental variables.
<code>qlegv.maps</code>	by default is set to NULL. Vector of strings, giving names of the raster maps, containing qualitative environmental variables (see details).
<code>presences.table</code>	data frame with species presence records in three columns, containing: <ul style="list-style-type: none"> • 1 - X coordinate or longitude. • 2 - Y coordinate or latitude. • 3 - number of presences or observations.
<code>col.w</code>	vector with column weights with same length as number of quantitative and qualitative maps.
<code>scannf</code>	logical. Whether number of factors should be introduced by user (TRUE) or not (FALSE). A barplot with eigenvalues after analysis is displayed.
<code>nf</code>	integer, indicating number of kept specialization axes when <code>scannf=FALSE</code> .
<code>method</code>	string character. Selection of method to perform the calculations, "normal" for normal data sets or "large" for large data sets with huge amount of data. See also details.
<code>load.maps</code>	logical. Whether produced maps should be uploaded into R. Note that if you are working with large data sets (<code>method="large"</code>), this option should be FALSE in order to avoid memory allocation problems.
<code>species.name</code>	string, indicating name of modelled species.
<code>map.center</code>	string, indicating name of a map for centering studied area (also a mask). When NULL, first EGV map will be used.
<code>res</code>	integer, indicating resolution for modelling. Unit should be the same as in used maps. When NULL, resolution will be fixed from <code>map.center</code> or first EGV map.

Details

Function `enirg` performs the Ecological Niche Factor Analysis (ENFA), following Hirzel *et al.* (2002). Analysis can be carried out for both quantitative variables and qualitative variables. This function requires occurrences records of the modelled species and environmental predictors of the study area.

User can choose number of factors that better describes the variance and the eigenvalue computation for factor extraction, by setting `scannf=TRUE`.

Depending on extension of the study area and environmental variables used to performed an analysis, user can choose one of two available methods: "normal" and "large". The first, strongly relies on the `rgrass7` package and thus is limited by R storage capacity and computations; on the contrary, "large" method directly interface with GRASS, allowing calculations over large areas or high resolution maps with huge amount of data (NOTE: it is only available for Linux/Unix OS, at the moment).

Global marginality is calculated as:

$$M = \frac{\sqrt{\sum_{i=1}^V m_i^2}}{1.96}$$

where m_i represents marginality of focal species on each EGV (up to V number of EGVs), in units of standard deviations of global distribution.

Global specialization is calculated as:

$$S = \frac{\sqrt{\sum_{i=1}^V \lambda_i}}{V}$$

where the eigenvalue λ_i associated to any factor expresses the amount of specialization it accounts for, i.e., ratio of the variance of global distribution to that of the species distribution on this axis.

If `load.maps` is set to `TRUE`, then maps of marginality and `nf` number of specialization maps will be loaded into R, by using raster library. Note that this is not recommended for large data sets.

Value

`enirg` returns a list object of class "enirg" containing the following components:

- `call`. Original call
- `nf`. An integer indicating the number of kept specialization axes
- `cw`. Column weights, a vector with p components
- `species`. Species' name
- `egvs`. A list of used environmental predictors
- `qt.egvs`. A list of used quantitative environmental predictors
- `ql.egvs`. A list of used quantitative environmental predictors
- `presences`. A data frame containing the observations
- `total.marginality`. Global marginality
- `marginalities`. Vector of marginalities for axis
- `total.specialization`. Global specialization
- `specializations`. Vector of specializations by axis
- `co`. Column coordinates, data frame with V rows and `nf` columns
- `mar`. Coordinates of marginality vector
- `m`. Marginality
- `s`. Vector of eigenvalues
- `obs.li`. Marginality and specialization axis for observations

Author(s)

Chiara Magliozzi <chiara.magliozzi@libero.it>, Fernando Canovas <fcgarcia@ualg.pt>, Jose Antonio Palazon-Ferrando <palazon@um.es>

References

Basille, M., Calenge, C., Marboutin, E., Andersen, R. & Gaillard, J.M. (2008). Assessing habitat selection using multivariate statistics: Some refinements of the ecological-niche factor analysis. *Ecological Modelling*, 211, 233-240.

Canovas, F., Magliozzi, C., Mestre, F., Palazon-Ferrando, J.A. and Gonzalez-Wanguemert, M. (2016). ENiRG: R-GRASS interface for efficiently characterizing the ecological niche of species and predicting habitat suitability. *Ecography* 39: 593-598.

Hirzel, A.H., Hausser, J., Chessel, D. & Perrin, N. (2002). Ecological-niche factor analysis: How to compute habitat-suitability maps without absence data? *Ecology*, 83, 2027-2036.

See Also

[stdz.maps](#), [enfa](#), [initGRASS](#)

Examples

```
## Not run:
# starting GRASS session
initGRASS("/usr/bin/grass-7.0.0", home=tempdir())
initGRASS("C:/GRASS", home=tempdir())

data(apis.enirg)

# presences table
lina <- apis.enirg$presences

# loading the environmental information in batch
predictor.names <- c("tann", "mntcm", "rfdm", "rfwm")
predictor.maps <- paste("std_", predictor.names, sep="")
file.names <- paste(system.file(package = "ENiRG"),
                    "/ext/", predictor.names, ".asc", sep="")

import.egvs(file.names, predictor.names)

# standardization
stdz.maps(predictor.names, predictor.maps)

# setting the mask
mask.file <- paste(system.file(package = "ENiRG"),
                  "/ext/", "mask.asc", sep="")
import.egvs(mask.file, "mask")

# performing the Ecological Niche Factor Analysis (ENFA)
enirg(presences.table = lina, qtegv.maps = predictor.maps,
      species.name = "African", nf = 3,
      scannf = FALSE, load.maps = TRUE
      map.center = "mask", method = "normal") -> apis.enirg

summary(apis.enirg)

require(raster)
```

```
plot(apis.enirg$African_li_Mar)
plot(apis.enirg$African_li_Spec1)

## End(Not run)
```

enirg.GUI

Graphical User Interface of the ENiRG package

Description

enirg.GUI creates a friendly GUI to set functions available in this package. Interface relies on *tcltk* and *fgui* routines to parse functions and create widgets.

Usage

```
enirg.GUI()
```

Value

Displays a graphic user interface.

Author(s)

Fernando Canovas <fcgarcia@ualg.pt>, Chiara Magliozzi <chiara.magliozzi@libero.it>

Examples

```
enirg.GUI()
```

enirg.plot

Ordination plot of [enirg](#) results

Description

Performs an ordination diagram for objects of class [enirg](#).

Usage

```
enirg.plot(enirg.results, mar.col = "grey", spe.col = "black",
           method = "extended", plot.egvs = TRUE, asp = FALSE,
           title = NULL)
```

Arguments

<code>enirg.results</code>	an object of class <code>enirg</code>
<code>mar.col</code>	color for filling available space (habitat)
<code>spe.col</code>	color for filling used space (presences data)
<code>method</code>	a string character. "extended" represents available space by using the convex hull, "simplified" represents available space by a polygon, representing the range.
<code>plot.egvs</code>	logical. If TRUE, ecogeographical variables will be also plotted in the ordination diagram.
<code>asp</code>	logical. If TRUE, marginality and specialization axes will be scaled.
<code>title</code>	string. A title for plot.

Details

`enirg.plot` displays the factorial map of pixels, by projection of the row coordinates of ENFA analysis. Available and used habitat are displayed as polygon area and clouds of points, respectively. Marginality axis is the X-axis and specialization is the Y-axis. Quantitative environmental predictors used for the analysis are represented using blue arrows and qualitative ones are represented by text labels. White dot shows the centroid of the used habitat.

Author(s)

Fernando Canovas <fcgarcia@ualg.pt>, Chiara Magliozzi <chiara.magliozzi@libero.it>

References

Basille, M., Calenge, C., Marboutin, E., Andersen, R. and Gaillard, J.M. (2008) Assessing habitat selection using multivariate statistics: Some refinements of the ecological-niche factor analysis. *Ecological Modelling*, 211, 233–240.

See Also

[enirg](#), [import.egvs](#), [enfa](#), [scatter.enfa](#), [initGRASS](#)

Examples

```
## Not run:
# starting GRASS session
initGRASS("/usr/bin/grass-7.0.0", home=tempdir())
initGRASS("C:/GRASS", home=tempdir())

data(apis.enirg)

# presences table
lina <- apis.enirg$presences

# loading the environmental information in batch
predictor.names <- c("tann", "mxtwm", "mntcm", "rfdm", "rfseas")
```

```

predictor.maps <- paste("std_", predictor.names, sep="")
file.names <- paste(system.file(package = "ENiRG"),
                    "/ext/", predictor.names, ".asc", sep="")

import.egvs(file.names, predictor.names)

# standardization
stdz.maps(predictor.names, predictor.maps)

# performing the Ecological Niche Factor Analysis (ENFA)
enirg(presences.table = lina, qtegv.maps = predictor.maps,
      species.name = "African", nf = 3, scannf = FALSE,
      method = "normal") -> apis.enfa

enirg.plot(apis.enfa)

## End(Not run)

```

enirg.predict

Predicting species' suitable habitat

Description

enirg.predict computes habitat suitability maps (HSM) using the Ecological Niche Factor Analysis (see [enirg](#)), and Mahalanobis distances method.

Usage

```

enirg.predict(enirg.results, qtegv.maps = NULL, qllegv.maps = NULL,
              load.map = FALSE, method = "normal",
              prediction.name = "predicted")

```

Arguments

enirg.results	object of class enirg
qtegv.maps	vector giving names of quantitative environmental variables raster maps. If set to NULL, automatically uses the same variables as the one used to perform enirg .
qllegv.maps	by default is set to NULL. vector of strings, giving names of raster maps, containing qualitative environmental variables (see details). If set to NULL, automatically uses the same variables as the one used to perform enirg .
load.map	logical. Whether map should be uploaded as an object of class raster .
method	string. "normal" or "large".
prediction.name	string. A suffix for naming derived maps.

Details

Function `enirg.predict`, bases prediction on the results obtained from `enirg` function. User can choose to upload new environmental variables (i.e., to make predictions under different scenarios), or to use the same predictors, which were provided to compute the ENFA analysis. In both cases position of the niche is determined computing Mahalanobis distances for each pixel from the barycentre of the niche using the row coordinates of the ENFA analysis. Computed HSM has values ranging from 0 (complete absence) to 1 (complete presence).

Results should be later classified by using `boyce`.

Depending on the extension of the study area and the environmental variables used to performed the analysis, User can choose one of two available methods: "normal" and "large". The first, strongly relies on the `rgrass7` package and thus is limited by R storage capacity and computations; on the contrary, "large" method directly interface with GRASS, allowing calculations over large areas or high resolution maps with huge amount of data, but it is only available for Linux/Unix OS, at the moment.

Value

It computes HSM and stores it as raster a map layer in GRASS. It also returns a list of:

- `predictions`. A data frame with observed and predicted values.
- `validation`. A data frame with distribution of predicted values for both the observed data and the entire predicted map.
- `map`. Prediction map can be also uploaded into R by setting `load.map` to `TRUE`

Author(s)

Fernando Canovas <fcgarcia@ualg.pt>, Chiara Magliozzi <chiara.magliozzi@libero.it>

References

Hirzel, A.H., Hausser, J., Chessel, D. \& Perrin, N. (2002) Ecological-niche factor analysis: How to compute habitat-suitability maps without absence data? *Ecology*, 83, 2027-2036.

See Also

[enirg](#), [import.egvs](#), [predict.enfa](#), [initGRASS](#)

Examples

```
## Not run:
# starting GRASS session
initGRASS("/usr/bin/grass-7.0.0", home=tempdir())
initGRASS("C:/GRASS", home=tempdir())

data(apis.enirg)

# presences table
lina <- apis.enirg$presences
```

```

# loading the environmental information in batch
predictor.names <- c("tann", "mxtwm", "mntcm", "rfdm", "rfseas")
predictor.maps <- paste("std_", predictor.names, sep="")
file.names <- paste(system.file(package = "ENiRG"),
                    "/ext/", predictor.names, ".asc", sep="")

import.egvs(file.names, predictor.names)

# standardization
stdz.maps(predictor.names, predictor.maps)

# performing the Ecological Niche Factor Analysis (ENFA)
enirg(presences.table = lina, qtegv.maps = predictor.maps,
      species.name = "African", nf = 1, scannf = FALSE,
      method = "normal") -> apis.enfa

enirg.predict(apis.enfa, load.map = TRUE, method = "normal") -> apis.hsm

## End(Not run)

require(raster)

# Results can be directly loaded from:
data(apis.hsm)

plot(apis.hsm$African_predicted_hsm)
contour(apis.hsm$African_predicted_hsm, add = TRUE)

```

import.egvs

Raster file importation

Description

This function allows to import raster files into raster maps layers supported in GRASS. Available formats are those provided by GDAL library (http://www.gdal.org/formats_list.html).

Usage

```
import.egvs(filenamees, output.names)
```

Arguments

filenamees	character string. Path to raster file to be imported.
output.names	character string. Name of output raster map that will be created in GRASS environment.

Details

Datum and projection of source dataset must match projection of current GRASS location, otherwise a report error message can be generated.

Value

A raster map layer will be uploaded into the opened GRASS environment.

Author(s)

Chiara Magliozzi <chiara.magliozzi@libero.it>, Fernando Canovas <fcgarcia@ualg.pt>

References

<http://grass.osgeo.org/grass64/manuals/r.in.gdal.html>

See Also

[list.maps](#), [initGRASS](#)

Examples

```
## Not run:
# starting GRASS session
initGRASS("/usr/bin/grass-7.0.0", home=tempdir())
initGRASS("C:/GRASS", home=tempdir())

## Path to file
file1 <- paste(system.file(package = "ENiRG"), "/ext/tann.asc", sep="")

import.egvs(file1, "tann")

list.maps()

## End(Not run)
```

list.maps

List available raster and/or vector maps

Description

This function gives a list of maps in the current GRASS session, matching with a user-specified pattern.

Usage

```
list.maps(prefix = "*")
```

Arguments

`prefix` character string. It states a pattern to search for.

Details

Argument `prefix` allows to list all maps matching a regular expression or pattern, i.e., "std_*" will search for all maps starting with "std_". See also https://en.wikipedia.org/wiki/Regular_expression regular expressions (aka regex) - from Wikipedia, the free encyclopedia.

Value

Return a list of all both vectorial and raster maps available in the current GRASS session, matching a regular expression.

Author(s)

Chiara Magliozzi <chiara.magliozzi@libero.it>, Fernando Canovas <fcgarcia@ualg.pt>

References

<https://grass.osgeo.org/grass70/manuals/g.list.html>

See Also

[map.info](#), [import.egvs](#), [initGRASS](#)

Examples

```
## Not run:
# starting GRASS session
initGRASS("/usr/bin/grass-7.0.0", home=tempdir())
initGRASS("C:/GRASS", home=tempdir())

## Path of the files to be read
predictor.names <- c("tann", "mntcm", "mxtwm", "rfann", "rfdm", "rfwm")
file.names <- paste(system.file(package = "ENiRG"),
                    "/ext/", predictor.names, ".asc", sep="")

import.egvs(file.names, predictor.names)

list.maps("rf*")
list.maps("*ann")

## End(Not run)
```

map.info	<i>Retrieve metadata of a raster file from current GRASS location.</i>
----------	--

Description

Metadata about GRASS environment and about a map (i.e., raster map resolution, type and range) will be returned as a list. User can choose GRASS format to store the information, i.e. in a .txt file, or to have output as a list in R.

Usage

```
map.info(map.name, format = "grass")
```

Arguments

map.name	character string. Name of a raster map stored in GRASS as returned from list.maps .
format	character string, by default "grass". Preferred output format between GRASS output format ("grass") and R list ("r").

Value

Return either a list or text.

Author(s)

Chiara Magliozzi <chiara.magliozzi@libero.it>, Fernando Canovas <fcgarcia@ualg.pt>

References

<https://grass.osgeo.org/grass70/manuals/r.info.html>

See Also

[import.egvs](#), [initGRASS](#)

Examples

```
## Not run:
# starting GRASS session
initGRASS("/usr/bin/grass-7.0.0", home=tempdir())
initGRASS("C:/GRASS", home=tempdir())

## Path to file
file1 <- paste(system.file(package = "ENiRG"),"/ext/tann.asc", sep="")
import.egvs(file1, "tann")

map.info("tann", format="r")
```

```
## End(Not run)
```

stdz.maps	<i>Standardization of raster maps</i>
-----------	---------------------------------------

Description

This function standardizes according to the mean of the values contained in a raster map, previously imported into GRASS.

Usage

```
stdz.maps(map.names, output.names = NULL)
```

Arguments

`map.names` character string. Name of a raster maps to be standardized.
`output.names` character string. Output name of the standardized raster map.

Details

When working with multiple rasters, it is often necessary to measure them at the same numerical scale. The standardization allows to easier find maximum likelihood estimates and thus compare one raster to the others. It is important to retrieve the mean and standard deviation for each raster individually and then normalize all the different raster maps. This function calculates univariate statistics (minimum and maximum cell values, range, arithmetic mean, population variance, standard deviation, and coefficient of variation) from the non-null cells of a raster map.

Normalized raster map is then calculated following this normalized equation:

$$Z_{(i,j)} = \frac{x_{(i,j)} - \bar{X}}{\sigma_X}$$

, where $Z_{(i,j)}$ is the normalized value in a particular cell of cartesian coordinates (i,j) in the resolution of the map, $x_{(i,j)}$ is the current value for that cell, \bar{X} is the mean of the map and σ_X is the standard deviation of the map.

Value

This function returns a new raster map resulted from the normalization. When no output names are provided, obtained maps will be the same as the input, but adding a prefix "std_"

Author(s)

Chiara Magliozzi <chiara.magliozzi@libero.it>, Fernando Canovas <fcgarcia@ualg.pt>

See Also

[import.egvs](#), [enirg](#), [initGRASS](#)

Examples

```
## Not run:
# starting GRASS session
# initGRASS("/usr/bin/grass-7.0.0", home=tempdir())
# initGRASS("C:/GRASS", home=tempdir())

## Path to the file to be imported
file1 <- paste(system.file(package = "ENiRG"), "/ext/tann.asc", sep="")

import.egvs(file1, "tann")

stdz.maps("tann", "std_tann")

## End(Not run)
```

Index

- *Topic **ENFA**
 - enirg, 10
 - ENiRG-package, 2
 - enirg.plot, 14
- *Topic **ENiRG**
 - ENiRG-package, 2
- *Topic **GRASS**
 - enirg, 10
 - list.maps, 19
- *Topic **HSM**
 - boyce, 6
 - classify.map, 9
 - enirg.predict, 16
- *Topic **classification**
 - boyce, 6
- *Topic **datasets**
 - apis.enirg, 4
 - apis.hsm, 5
 - cbi.results, 8
- *Topic **enirg**
 - enirg.GUI, 14
 - enirg.plot, 14
- *Topic **gui**
 - enirg.GUI, 14
- *Topic **map**
 - import.egvs, 18
 - map.info, 21
- *Topic **metadata**
 - list.maps, 19
 - map.info, 21
- *Topic **niche**
 - enirg, 10
- *Topic **ordination**
 - enirg.plot, 14
- *Topic **prediction**
 - enirg.predict, 16
- *Topic **predict**
 - classify.map, 9
- *Topic **raster**
 - import.egvs, 18
 - list.maps, 19
 - map.info, 21
 - stdz.maps, 22
- *Topic **standardization**
 - stdz.maps, 22
- *Topic **vector**
 - list.maps, 19
- apis.enirg, 4, 8
- apis.hsm, 5
- boyce, 2, 3, 6, 8, 9, 17
- cbi.results, 8
- classify.map, 2, 3, 6, 7, 9
- enfa, 13, 15
- enirg, 2, 4, 9, 10, 14–17, 22
- ENiRG-package, 2
- enirg.GUI, 14
- enirg.plot, 2, 14
- enirg.predict, 2, 3, 5–7, 9, 16
- import.egvs, 2, 15, 17, 18, 20–22
- initGRASS, 2, 13, 15, 17, 19–22
- list.maps, 2, 19, 19, 21
- map.info, 2, 20, 21
- predict.enfa, 7, 17
- raster, 9, 16
- scatter.enfa, 15
- stdz.maps, 2, 13, 22