

Package ‘ENiRG’

February 19, 2015

Type Package

Title Ecological Niche in R and GRASS

Version 0.1

Date 2014-05-21

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

Maintainer Fernando Garcia Canovas <fcgarcia@ualg.pt>

Description

The package allows to perform the Ecological Niche Factor Analysis, calculate habitat suitability maps and classify the habitat in suitability classes. Computations are executed in a throw-away GRASS environment from R in order to be able to perform analysis with large data sets.

License GPL (>= 2)

Depends ade4, raster, R.utils, sp, sgrass6

Imports gdata, fgui, miniGUI, tcltk

NeedsCompilation no

Repository CRAN

Date/Publication 2014-10-09 14:33:32

R topics documented:

ENiRG-package	2
apis.enirg	4
apis.hsm	4
boyce	5
cbi.results	7
classify.map	8
enirg	10
enirg.GUI	13
enirg.plot	13
enirg.predict	15
import.egvs	17
list.maps	18

map.info	19
stdz.maps	20

Index	23
--------------	-----------

ENiRG-package *Ecological Niche in R-Grass*

Description

The package 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 the `spgrass6` package to deal with spatial data and to interface R and Grass.

Details

Package: ENiRG
 Type: Package
 Version: 0.1_1
 Date: 2014-06-02
 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 the function `initGRASS` from `spgrass6`). *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.

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

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

The main functions of the package offer two methods, accordingly to the amount of data computed. The user can choose one of the two available methods: "normal" and "large". The first, strongly relies on the `spgrass6` package and thus is limited by R storage capacity and computations; on the

contrary the "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 the function `enirg.predict` is predicting the modelled species presences (Boyce *et al.*, 2002). The intervals estimated manually, allow the user to reclassify the predicted niche map, by using `classify.map`, and thus distinguishing unsuitable, marginal, suitable and optimal habitat (Hirzel *et al.*, 2006). Evaluation of the habitat suitability model accuracy is made by means of 10-fold cross-validation (Fielding and Bell, 1997).

The user can take advantages of the integration with other R libraries (*raster*, *rasterVIS*), the 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 the functionalities of *ENiRG* package through the function `link{enirg.GUI}`.

Note

The package depends on `spgrass6`, `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

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.

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.

See Also

`adehabitat`, `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*. DOI: 10.1007/s10841-014-9701-1.

Examples

```
data(apis.enirg)
plot(apis.enirg$presences[, 1:2],
     cex=apis.enirg$presences[, 3] / 10, pch=19)
```

apis.hsm *List with values returned from enirg.predict.*

Description

apis.hsm is a list which stores the output of the 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 Rua, 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 Rua, 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 Rua, 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*. DOI: 10.1007/s10841-014-9701-1.

Examples

```
data(apis.hsm)

require(raster)

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

boyce

Habitat suitability classification using continuous Boyce index (CBI).

Description

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

The curves from the 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 the predicted-to-expected ratio (P/E) by partitioning the habitat suitability predictions into classes and by calculating frequencies. If the model properly delineates the suitable areas for the studied species, the Spearman rank correlation coefficient of the ratio F_i , will be 1.

Evaluation of the habitat suitability model accuracy is made by means of 10-fold cross-validation (Fielding & Bell, 1997), partitioning the data evenly but randomly into `cv.sets` partitions, defined by the user. Once the ratio is calculated, Spearman correlation coefficient allows to estimate the fitting for the predicted-to-expected relationship.

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

Usage

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

Arguments

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

Details

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

Value

The function displays the predicted/expected curve shapes. `Asol enirg` 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_hsm@data@values))

# Try with intervals:
# unsuitable = 0.25
# marginal = 0.69
# unsuitable = 0.81

boyce(prediction = apis.predictions,
       prediction.map = apis.predictions.map,
       categories = c(0, 0.25, 0.69, 0.81, 1),
       cv.sets = 10, type = "none")
```

cbi.results

List with values returned from [boyce](#).

Description

cbi.results is a list which stores the output of the continuous Boyce classification performed with [boyce](#).

Usage

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

Details

The source dataset refers to the 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_hsm@data@values))

# Try with intervals:
# unsuitable = 0.25
# marginal = 0.69
# unsuitable = 0.81

boyce(prediction = apis.predictions,
       prediction.map = apis.predictions.map,
       categories = c(0, 0.25, 0.69, 0.81, 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(enirg.results, suit.classes, output.name, load.map = FALSE)
```

Arguments

enirg.results	object of class enirg
suit.classes	object of class boyce , giving the suitability classes.
output.name	string. Name for the classified map. If set to NULL (by default), suffix "_hsm_class" will be automatically added to the species name stored into the enirg object.
load.map	logical. Whether the map should be uploaded as an object of class raster

Value

The function `classify.map` allows to classify the HSM according to the breaks, which are the limits for the 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, which 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-6.4.0", 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", "rfann", "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)

# 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

data(cbi.results)

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

## End(Not run)
```

enirg

*Ecological Niche in R-Grass***Description**

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

Usage

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

Arguments

qtegv.maps	vector of strings, giving the names of the raster maps, containing quantitative environmental variables.
qlegv.maps	by default is set to NULL. vector of strings, giving the names of the raster maps, containing qualitative environmental variables (see details).
presences.table	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 of observations.
col.w	vector with column weights with the same length as the number of quantitative and qualitative maps.
scannf	logical. Whether the number of factors should be introduced by the user (TRUE) or not (FALSE). A barplot with the eigenvalues of the analysis is displayed.
nf	integer, indicating the number of kept specialization axes when scannf=FALSE.
method	string character. Selection of the method to perform the calculations, "normal" for normal data sets or "large" for large data sets with huge amount of data. See also details.
load.maps	logical. Whether the produced maps should be uploaded into R. Note that if you are working with large data sets (method="large"), this option should be FALSE in order to avoid memory allocation problems.
species.name	string, indicating the name of the modelled species.

Details

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

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

Depending on the extension of the study area and the environmental variables used to performed the analysis, the user can choose one of the two available methods: "normal" and "large". The first, strongly relies on the `spgrass6` package and thus is limited by R storage capacity and computations; on the contrary the "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 the marginality of the focal species on each EGV (up to V number of EGVs), in units of standards deviations of the global distribution.

Global specialization is calculated as:

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

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

If `load.maps` is set to `TRUE`, then the 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 the used environmental predictors
- `qt.egvs`. A list of the used quantitative environmental predictors
- `ql.egvs`. A list of the used quantitative environmental predictors
- `presences`. A data frame with the table of 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 the marginality vector
- `m`. Marginality
- `s`. Vector of the 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.

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-6.4.0", 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", "rfann", "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)

# 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.plot(apis.enirg, method = "simplified")

## End(Not run)
```

`enirg.GUI`*Graphical User Interface of the ENiRG package*

Description

`enirg.GUI` creates a friendly GUI interface to set the functions available in the package. The interface relies on the *tcltk* and *fgui* routines to parse the functions and create widgets.

Usage

```
enirg.GUI()
```

Value

Displays the graphic user interface.

Author(s)

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

Examples

```
## Not run:
# starting GRASS session
initGRASS("/usr/bin/grass-6.4.0", home=tempdir())
initGRASS("C:/GRASS", home=tempdir())

## End(Not run)

enirg.GUI()
```

`enirg.plot`*Ordination plot of `enirg` results*

Description

Performs the ordination diagram for objects of class `enirg`.

Usage

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

Arguments

enirg.results	an object of class enirg
mar.col	the color for filling the available space (habitat)
spe.col	the color for filling the used space (presences data)
method	a string character. "Extended" represents a polygon of the available space, "simplified" represents a clouds of points for representing the used space, with size proportional to the abundance.
plot.egvs	logical. If TRUE, ecogeographical variables will be also plotted in the ordination diagram.

Details

enirg.plot displays the factorial map of pixels, by projection of the row coordinates of ENFA analysis. The available and the used habitat are displayed as polygon area or clouds of points. The marginality axis is the X-axis and the specialization is the Y-axis. The quantitative environmental predictors used for the analysis are represented using blue arrows and qualitative ones are represented by text labels.

Author(s)

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

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.

See Also

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

Examples

```
## Not run:
# starting GRASS session
initGRASS("/usr/bin/grass-6.4.0", 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", "rfann", "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)

# 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.plot(apis.enirg, method = "simplified")

## End(Not run)

```

enirg.predict

Predicting species' suitable habitat

Description

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

Usage

```

enirg.predict(enirg.results, qtegv.maps = NULL, qlegv.maps = NULL,
              load.map = FALSE, method = "normal")

```

Arguments

enirg.results	object of class enirg
qtegv.maps	vector giving the names of the quantitative environmental variables raster maps. If set to NULL, automatically uses the same variables as the one used to perform enirg .
qlegv.maps	by default is set to NULL. vector of strings, giving the names of the 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 the map should be uploaded as an object of class raster
method	string. "normal" or "large"

Details

The function `enirg.predict`, bases the prediction on the results obtained from `enirg` function. The 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 the position of the niche is determined computing the Mahalanobis distances for

each pixel from the barycentre of the niche using the row coordinates of the ENFA analysis. The computed HSM has values ranging from 0 (complete absence) to 1 (complete presence).

The 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, the user can choose one of the two available methods: "normal" and "large". The first, strongly relies on the `spgrass6` package and thus is limited by R storage capacity and computations; on the contrary the "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 the 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 the 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-6.4.0", 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", "rfann", "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)

# 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)

data(apis.hsm)

plot(apis.hsm$African_hsm)
contour(apis.hsm$African_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. The name of the raster file to be imported.
`output.names` character string. The name of the output raster map that will be created.

Details

Datum and projection of the source dataset must match the projection of the current 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-6.4.0", home=tempdir())
initGRASS("C:/GRASS", home=tempdir())

## Path of the file to be imported
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

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

Usage

```
list.maps(type = c("rast", "vect"), prefix = "*")
```

Arguments

type character string from "rast" (raster maps), or "vect" (vectorial maps).
 prefix character string. It states the pattern to search for.

Details

The argument prefix allows to list all maps matching a regular expression or pattern, i.e., "std_*" will search for all the maps starting with "std_".

Value

Return a vector of all the maps available in the current GRASS session or matching a regular expression.

Author(s)

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

References

<http://grass.osgeo.org/grass65/manuals/g.mlist.html>

See Also

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

Examples

```
## Not run:
# starting GRASS session
initGRASS("/usr/bin/grass-6.4.0", home=tempdir())
initGRASS("C:/GRASS", home=tempdir())

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

list.maps("rast")

## End(Not run)
```

map.info

Retrieve metadata of a raster file from the current GRASS location.

Description

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

Usage

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

Arguments

map.name	character string. The name of the raster map stored in GRASS as returned from list.maps .
format	character string, by default "grass". The 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

<http://grass.osgeo.org/grass65/manuals/r.info.html>

See Also

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

Examples

```
## Not run:
# starting GRASS session
initGRASS("/usr/bin/grass-6.4.0", home=tempdir())
initGRASS("C:/GRASS", home=tempdir())

## Path of the file to be read if we want just one of the variables
file1 <- paste(system.file(package = "ENiRG"),"/ext/tann.asc", sep="")
import.egvs(file1, "tann")

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

## End(Not run)
```

stdz.maps

Standardization of raster maps

Description

This function standardize 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. The name of the raster maps to be standardized.
`output.names` character string. The 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 the 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. The function calculates the 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.

The normalized raster map is calculated following the 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 the cell, \bar{X} is the mean of the map and σ_X is the standard deviation of the map.

Value

The function returns a new raster map resulted from the normalization with the arithmetic expression. When no output names are provided, the 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>

References

<http://grass.osgeo.org/grass65/manuals/r.univar.html> <http://grass.osgeo.org/grass64/manuals/r.mapcalc.html>

See Also

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

Examples

```
## Not run:
# starting GRASS session
# initGRASS("/usr/bin/grass-6.4.0", home=tempdir())
# initGRASS("C:/GRASS", home=tempdir())

## Path of 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, 13
- *Topic **ENiRG**
 - ENiRG-package, 2
- *Topic **GRASS**
 - enirg, 10
 - list.maps, 18
- *Topic **HSM**
 - boyce, 5
 - classify.map, 8
 - enirg.predict, 15
- *Topic **classification**
 - boyce, 5
- *Topic **datasets**
 - apis.enirg, 4
 - apis.hsm, 4
 - cbi.results, 7
- *Topic **enirg**
 - enirg.GUI, 13
 - enirg.plot, 13
- *Topic **gui**
 - enirg.GUI, 13
- *Topic **map**
 - import.egvs, 17
 - map.info, 19
- *Topic **metadata**
 - list.maps, 18
 - map.info, 19
- *Topic **niche**
 - enirg, 10
- *Topic **ordination**
 - enirg.plot, 13
- *Topic **prediction**
 - enirg.predict, 15
- *Topic **predict**
 - classify.map, 8
- *Topic **raster**
 - import.egvs, 17
 - list.maps, 18
 - map.info, 19
 - stdz.maps, 20
- *Topic **standardization**
 - stdz.maps, 20
- *Topic **vector**
 - list.maps, 18
- apis.enirg, 4, 7
- apis.hsm, 4
- boyce, 2, 3, 5, 7, 8, 16
- cbi.results, 7
- classify.map, 2, 3, 5, 6, 8
- enfa, 12, 14
- enirg, 2, 4, 8, 9, 10, 13–16, 21
- ENiRG-package, 2
- enirg.GUI, 13
- enirg.plot, 2, 13
- enirg.predict, 2–6, 9, 15
- import.egvs, 2, 14, 16, 17, 19–21
- initGRASS, 2, 12, 14, 16, 18–21
- list.maps, 2, 18, 18, 20
- map.info, 2, 19, 19
- predict.enfa, 6, 16
- raster, 8, 15
- scatter.enfa, 14
- stdz.maps, 2, 12, 20