Package ‘MIAmaxent’

September 14, 2016

Type Package

Title Maxent Distribution Model Selection

Version 0.3.7

Maintainer Julien Vollering <julien.vollering@hisf.no>

Description Tools for training, selecting, and evaluating maximum entropy (Maxent) distribution models. This package provides tools for user-controlled transformation of explanatory variables, selection of variables by nested model comparison, and flexible model evaluation and projection. It is based on the strict maximum likelihood interpretation of maximum entropy modelling.

Depends R (>= 2.10)

License MIT + file LICENSE

LazyData TRUE

URL https://github.com/julienvollering/MIAmaxent

BugReports https://github.com/julienvollering/MIAmaxent/issues

RoxygenNote 5.0.1

SystemRequirements Maxent.jar software, Java Runtime Environment (JRE)

Imports dplyr (>= 0.4.3), e1071 (>= 1.6-7), graphics, grDevices, Hmisc (>= 3.17-2), raster (>= 2.5-8), stats, utils

Suggests knitr, rmarkdown, R.rsp

VignetteBuilder R.rsp

NeedsCompilation no

Author Julien Vollering [aut, cre],
Sabrina Mazzoni [aut],
Rune Halvorsen [aut]

Repository CRAN

Date/Publication 2016-09-14 11:46:11

R topics documented:

`deriveVars` *Derive variables by transformation.*

Description

`deriveVars` produces derived variables from explanatory variables by transformation, and returns a list of dataframes. The available transformation types are as follows, described in Halvorsen et al. (2015): L, M, D, HF, HR, T (for continuous EVs), and B (for categorical EVs). For spline transformation types (HF, HR, T), a subset of possible DVs is selected by the criteria described under Details.

Usage

```
deriveVars(data, transformtype = c("L", "M", "D", "HF", "HR", "T", "B"),
  allsplines = FALSE, dir = NULL)
```

Arguments

`data`	Data frame containing the response variable in the first column and explanatory variables in subsequent columns. The response variable should represent presence/background data, coded as: 1/NA. See `readData`.
`transformtype`	Specifies the types of transformations types to be performed. Default is the full set of the following transfomation types: L (linear), M (monotonous), D (deviation), HF (forward hinge), HR (reverse hinge), T (threshold), and B (binary).
`allsplines`	Logical. Keep all spline transformations created, rather than selecting particular splines based on fraction of total variation explained.
`dir`	Directory to which files will be written during selection of spline-type derived variables. Defaults to the working directory.

Details

The linear transformation "L" is a simple rescaling to the range [0, 1].

The monotonous transformation "M" performed is a zero-skew transformation (Oekland et al. 2001).

The deviation transformation "D" is performed around an optimum EV value that is found by looking at frequency of presence (see `plotFOP`). Three deviation transformations are created with different steepness and curvature around the optimum.

For spline transformations ("HF", "HR", and "T"), DVs are created around 20 different break points (knots) which span the range of the EV. Only DVs which satisfy all of the following criteria are retained:

1. 3 <= knot <= 18 (DVs with knots at the extremes of the EV are never retained).
2. F-test of the single-variable Maxent model from the given DV gives a p-value < 0.05.
3. The single-variable Maxent model from the given DV shows a local maximum in fraction of variation explained (FVA) compared to DVs from the neighboring 4 knots.

For categorical variables, 1 binary derived variable (type "B") is created for each category.

Explanatory variables should be uniquely named, and the names must not contain spaces, underscores, or colons. Underscores and colons are reserved to denote derived variables and interaction terms repectively.

Value

List of 2:

1. A list of data frames, with each containing the derived variables produced for a given explanatory variable. This item is recommended as input for `dvdata` in `selectDVforEV`.
2. A list of all the transformation functions used to produce the derived variables. This item is recommended as input for `transformation` in `plotResp2`, `testAUC`, and `projectModel`.

References

Halvorsen, R., Mazzoni, S., Bryn, A., & Bakkestuen, V. (2015). Opportunities for improved distribution modelling practice via a strict maximum likelihood interpretation of MaxEnt. Ecography, 38(2), 172-183.

Oekland, R.H., Oekland, T. & Rydgren, K. (2001). Vegetation-environment relationships of boreal spruce swamp forests in Oestmarka Nature Reserve, SE Norway. Sommerfeltia, 29, 1-190.

Examples

```
## Not run:
deriveddat <- deriveVars(dat, transformtype = c("HF", "HR", "T"), allsplines = TRUE,
   dir = "D:/path/to/modeling/directory")

## End(Not run)

toydata_dvs <- deriveVars(toydata_sp1po, transformtype = c("L", "M", "D", "B"))
str(toydata_dvs$EVDV)
```

```
summary(toydata_dvs$transformations)

## Not run:
# From vignette:
grasslandDVs <- deriveVars(grasslandPO, transformtype = c("L", "M", "D", "HF", "HR", "T", "B"))
summary(grasslandDVs$EVDV) # alternatively: summary(grasslandDVs[[1]])
head(summary(grasslandDVs$transformations)) # alternatively: head(summary(grasslandDVs[[2]]))
length(grasslandDVs$transformations)
plot(grasslandPO$terslpdg, grasslandDVs$EVDV$terslpdg$terslpdg_D2, pch = 20)
plot(grasslandPO$terslpdg, grasslandDVs$EVDV$terslpdg$terslpdg_HR4, pch = 20)

## End(Not run)
```

`MIAmaxent` *MIAmaxent: Maxent Distribution Model Selection.*

Description

Training, selecting, and evaluating maximum entropy (Maxent) distribution models. This package provides tools for user-controlled transformation of explanatory variables, selection of variables by nested model comparison, and flexible model evaluation and projection. The methods implemented here are based on the strict maximum likelihood interpretation of maximum entropy modelling (Halvorsen, 2013, Halvorsen et al., 2015). The predecessor to this package is the MIA Toolbox, which is described in detail in Mazzoni et al. (2015).

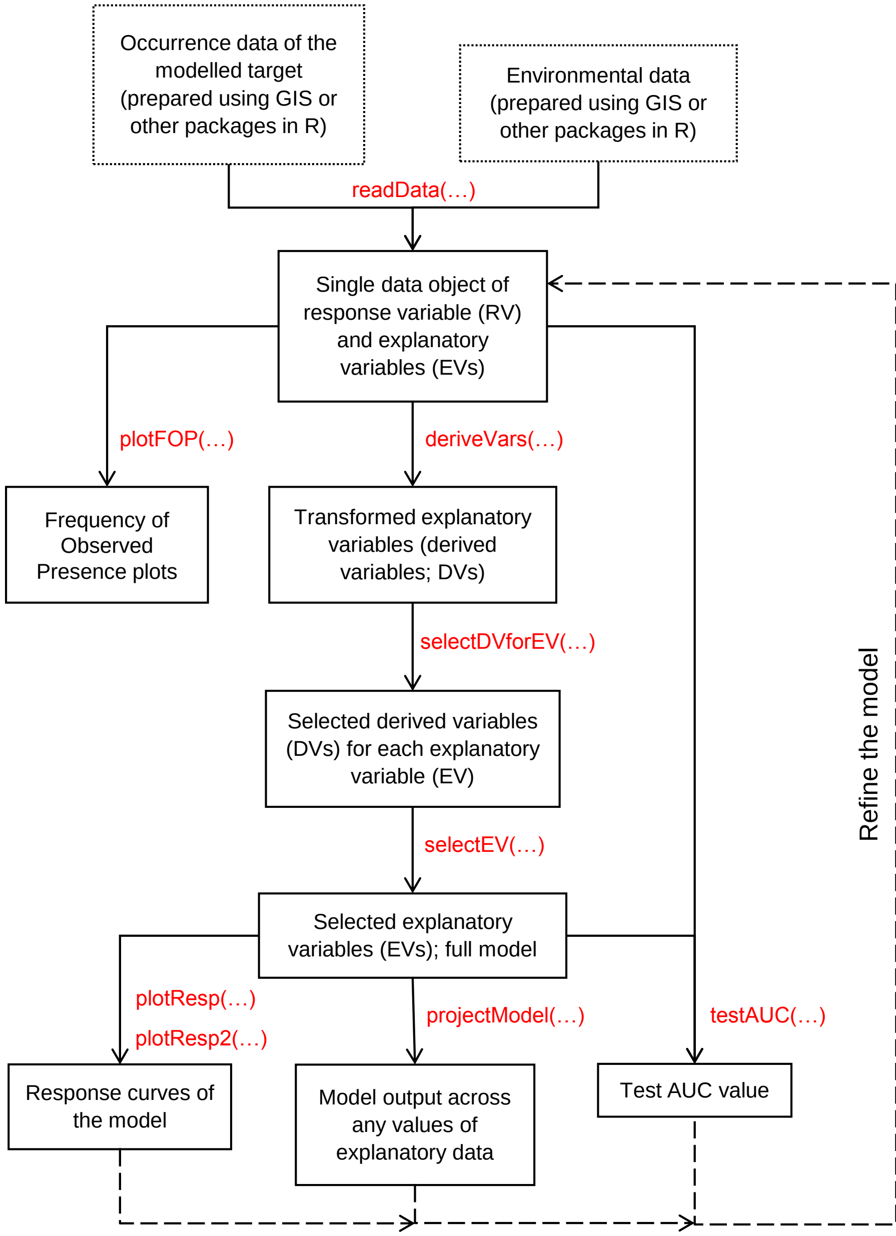
System Requirements

The maximum entropy algorithm utilized in this package is provided by the MaxEnt Java program (Phillips et al., 2006). This software is freely available, but may not be distributed further. Therefore, you must download the MaxEnt program (v3.3.3k) from https://www.cs.princeton.edu/~schapire/maxent/, and place the 'maxent.jar' file in the 'java' folder of this package. This folder can be located by the following R command: `system.file("java", package = "MIAmaxent")`.

You must have the Java Runtime Environment (JRE) installed on your computer for the MaxEnt program to function. You can check if you have Java installed, and download it if necessary, at http://java.com/download.

User Workflow

This diagram outlines a common workflow for users of this package. Functions are shown in red.



References

Halvorsen, R. (2013) A strict maximum likelihood explanation of MaxEnt, and some implications for distribution modelling. Sommerfeltia, 36, 1-132.

Halvorsen, R., Mazzoni, S., Bryn, A. & Bakkestuen, V. (2015) Opportunities for improved distribution modelling practice via a strict maximum likelihood interpretation of MaxEnt. Ecography, 38, 172-183.

Mazzoni, S., Halvorsen, R. & Bakkestuen, V. (2015) MIAT: Modular R-wrappers for flexible implementation of MaxEnt distribution modelling. Ecological Informatics, 30, 215-221.

Phillips, S.J., Anderson, R.P. & Schapire, R.E. (2006) Maximum entropy modeling of species geographic distributions. Ecological Modelling, 190, 231-259.

`plotFOP` *Plot Frequency of Observed Presence (FOP).*

Description

`plotFOP` produces a Frequency of Observed Presence (FOP) plot for a given explanatory variable. An FOP plot shows the rate of occurrence of the response variable across intervals or levels of the explanatory variable. For continuous variables, the exponentially weighted moving average of the FOP values is added to the plot as a line. `plotFOP` also returns a list containing the optimum EV value, and a data frame containing the plotted data (for customizable plotting).

Usage

```
plotFOP(data, EV, smoothwindow = 5, EVranging = FALSE, intervals = NULL,
  ...)
```

Arguments

`data` Data frame containing the response variable in the first column and explanatory variables in subsequent columns. The response variable should represent presence or background, coded as: 1/NA. See Details for information regarding presence/absence data. See also `readData`.

`EV` Name or column index of the explanatory variable in `data` for which to calculate FOP.

`smoothwindow` Width of the smoothing window. Represents the number of intervals included in an exponentially weighted moving average. Should be odd, otherwise the window will be uncentered. Irrelevant for categorical EVs.

`EVranging` Logical. If `TRUE`, will range the EV scale to [0,1]. This is equivalent to plotting FOP over the linear transformation produced by deriveVars. Irrelevant for categorical EVs.

`intervals` Number of intervals into which the continuous EV is divided. Defaults to the minimum of N/50 and 100. Irrelevant for categorical EVs.

`...` Arguments to be passed to `plot` to control the appearance of the plot. For example:

- `cex` for size of points
- `col` for color
- `xlim` for range of the x-axis

Details

If the response variable in `data` represents presence/absence data, the result is an empirical frequency of presence curve, rather than a observed frequency of presence curve (see Stoea et al. [in press], Sommerfeltia).

The returned value of 'EVoptimum' is based on the smoothed FOP values, such that an outlying maximum in FOP may, in some cases, not be considered the optimal value of EV.

Value

In addition to the graphical output, a list of 2:

1. `EVoptimum`. The EV value (or level, for categorical EVs) at which FOP is highest
2. `FOPdata`. A data frame containing the plotted data. Columns in this data frame represent the following: EV interval ("int"), number of points in the interval ("n"), mean EV value of the points in the interval ("intEV"), mean RV value of the points in the interval ("intRV"), and exponentially weighted moving average of intRV ("smoothRV"). For categorical variables, only the number of points in the level ("n"), the level name ("level"), and the mean RV value of the level ("levelRV") are used.

References

Stoea, B., Halvorsen, R., Mazzoni, S. & Gusarov, V. (2016) Sampling bias in presence-only data used for species distribution modelling: assessment and effects on models. Sommerfeltia, submitted manuscript.

Examples

```
FOPev11 <- plotFOP(toydata_sp1po, 2, intervals = 5, smoothwindow = 3)
FOPev11 <- plotFOP(toydata_sp1po, 2, intervals = 8)
FOPev12 <- plotFOP(toydata_sp1po, "EV12", intervals = 8, ylim=c(0,1))
FOPev12$EVoptimum
FOPev12$FOPdata

## Not run:
# From vignette:
teraspifFOP <- plotFOP(grasslandPO, "teraspif")
terslpdgFOP <- plotFOP(grasslandPO, "terslpdg", intervals = 25,  pch=20, cex=1.1, col = "red")
terslpdgFOP$EVoptimum
terslpdgFOP$FOPdata

## End(Not run)
```

plotResp — *Plot single-effect model response.*

Description

`plotResp` plots the single-effect response of a given Maxent model over any of the included explanatory variables (EVs) in that model. For categorical variables, a bar plot is returned rather than a scatter plot. `plotResp` also returns a data frame containing the plotted data (for customizable graphics). Single-effect response curves present the response of a model containing the explanatory variable of interest only (cf. marginal-effect response curves; `plotResp2`).

Usage

```
plotResp(data, dvdata, EV, dir = NULL, logscale = FALSE, ...)
```

Arguments

`data`	Data frame containing the response variable in the first column and explanatory variables in subsequent columns. The response variable should represent presence/background data, coded as: 1/NA. See `readData`.
`dvdata`	List of explanatory variables used to train the model, where each list item is a data frame containing *selected* DVs for a *selected* EV (e.g. the first item in the list returned by `selectEV`).
`EV`	Name or list index of the explanatory variable in `dvdata` for which the response curve is to be generated. Interaction terms not allowed.
`dir`	Directory to which files will be written. Defaults to the working directory.
`logscale`	Logical. Plot the common logarithm of PRO rather than PRO itself.
`...`	Arguments to be passed to `plot` to control the appearance of the plot. For example: • `cex` for size of points • `col` for color • `pch` for type

Details

The plot contains points, representing the model response across individual data points, as well as a line, representing an exponentially weighted moving average of the model response over intervals of the EV.

The `EV` specified in `dvdata` must not be an interaction term.

Value

In addition to the graphical output, the plotted data is returned. In the case of a continuous EV, the plotted data is a list of 2:

1. `respPts`. Model response across individual data points. Columns in this data frame represent the following: EV value ("EV"), Probability Ratio Output of the model ("PRO"), and corresponding EV interval ("int").
2. `respLine`. Model response across intervals of the EV. Columns in this data frame represent the following: EV interval ("int"), number of points in the interval ("n"), mean EV value of the points in the interval ("intEV"), mean Probability Ratio Output of the points in the interval ("intPRO"), and exponentially weighted moving average of intPRO ("smoothPRO").

In the case of a categorical EV, the plotted data is a data frame containing the number of points in the level ("n"), the level name ("level"), and the mean Probability Ratio Output of the level ("levelRV").

Examples

```
## Not run:
responseEV1 <- plotResp(dat, deriveddat, "EV1", dir = "D:/path/to/modeling/directory")

# From vignette
pr_bygallResp <- plotResp(grasslandPO, grasslandEVselect[[1]], "pr_bygall")
par(mfrow = c(1,2))
pr_bygallFOP <- plotFOP(grasslandPO, "pr_bygall", intervals=50)
pr_bygallResp <- plotResp(grasslandPO, grasslandEVselect[[1]], "pr_bygall")
par(mfrow = c(1,1))

## End(Not run)
```

---

plotResp2 *Plot marginal-effect model response.*

Description

`plotResp2` plots the marginal-effect response of a given Maxent model over any of the included explanatory variables (EVs) in that model. For categorical variables, a bar plot is returned rather than a scatter plot. `plotResp2` also returns a data frame containing the plotted data (for customizable graphics). Marginal-effect response curves present the response of the model when all other explanatory variables are held constant at their mean values (cf. single-effect response curves; `plotResp`).

Usage

```
plotResp2(data, EV, transformation, model, logscale = FALSE, ...)
```

Arguments

`data`	Data frame of explanatory variables (EVs) included in the model, with column names matching EV names. See `readData`.
`EV`	Name or column index of the explanatory variable in `data` for which the response curve is to be generated.
`transformation`	Full pathway of the 'transformations.Rdata' file containing the transformations used to build the model. This file is saved as a result of the `deriveVars` function. Equivalently, the second item in the list returned by `deriveVars` can be used directly.
`model`	Full pathway of the '.lambdas' file of the model in question. This file is saved as a result of `selectEV`.
`logscale`	Logical. Plot the common logarithm of PRO rather than PRO itself.
`...`	Arguments to be passed to `plot` to control the appearance of the plot. For example: • `cex` for size of points • `col` for color • `pch` for type

Details

The plot contains points, representing the model response across individual data points, as well as a line, representing an exponentially weighted moving average of the model response over intervals of the EV.

Model response is commonly plotted across EV values of the training data, but it is possible to plot the model response over any EV values supplied in `data`.

The `EV` specified in `data` must not be an interaction term.

Value

In addition to the graphical output, the plotted data is returned. In the case of a continuous EV, the plotted data is a list of 2:

1. `respPts`. Model response across individual data points. Columns in this data frame represent the following: EV value ("EV"), Probability Ratio Output of the model ("PRO"), and corresponding EV interval ("int").
2. `respLine`. Model response across intervals of the EV. Columns in this data frame represent the following: EV interval ("int"), number of points in the interval ("n"), mean EV value of the points in the interval ("intEV"), mean Probability Ratio Output of the points in the interval ("intPRO"), and exponentially weighted moving average of intPRO ("smoothPRO").

In the case of a categorical EV, the plotted data is a data frame containing the number of points in the level ("n"), the level name ("level"), and the mean Probability Ratio Output of the level ("levelRV").

Examples

```
## Not run:
responseEV1 <- plotResp2(dat, "EV1",
   transformation = "D:/path/to/modeling/directory/deriveVars/transformations.Rdata",
   model = "D:/path/to/modeling/directory/selectEV/round/model/1.lambdas")

## End(Not run)

names(toydata_selevs$selectedEV)
resp <- plotResp2(toydata_sp1po, "EV11", toydata_dvs$transformations,
   system.file("extdata/sommerfeltia", "1.lambdas", package = "MIAmaxent"))

## Not run:
# From vignette:
pr_bygallResp2 <- plotResp2(grasslandPO, "pr_bygall",
transformation = grasslandDVs[[2]],
model = system.file("extdata", "1.lambdas", package = "MIAmaxent"))

## End(Not run)
```

`projectModel` *Project model to data.*

Description

`projectModel` Calculates the probability ratio output (PRO) of a given model for any points where values of the explanatory variables in the model are known.The transformations performed on the explanatory variables to build the model must be specified.

Usage

```
projectModel(data, transformation, model, clamping = FALSE, rescale = FALSE,
  raw = FALSE)
```

Arguments

`data`	Data frame of all the explanatory variables (EVs) included in the model, with column names matching EV names. See `readData`.
`transformation`	Full pathway of the 'transformations.Rdata' file containing the transformations used to build the model. This file is saved as a result of the `deriveVars` function. Equivalently, the second item in the list returned by `deriveVars` can be used directly.
`model`	Full pathway of the '.lambdas' file of the model in question. This file is saved as a result of `selectEV`.
`clamping`	Logical. Do clamping *sensu* Phillips et al. (2006). Default is `FALSE`.

rescale — Logical. Linearly rescale model output (PRO or raw) with respect to the projection `data`? This has implications for the interpretation of output values with respect to reference values (e.g. PRO = 1). See details.

raw — Logical. Return raw Maxent output instead of probability ratio output (PRO)? Default is FALSE.

Details

Missing data (NA) for a continuous variable will result in NA output for that point. Missing data for a categorical variable is treated as belonging to none of the categories.

When `rescale = FALSE` the scale of the model output (PRO or raw) returned by this function is dependent on the data used to train the model. For example, a location with PRO = 2 can be interpreted as having a probability of presence twice as high as an average site in the *training* data (Halvorsen, 2013, Halvorsen et al., 2015). When `rescale = TRUE`, the output is linearly rescaled with respect to the data onto which the model is projected. In this case, a location with PRO = 2 can be interpreted as having a probability of presence twice as high as an average site in the *projection* data. Similarly, raw values are on a scale which is dependent on the size of either the training data extent (`rescale = FALSE`) or projection data extent (`rescale = TRUE`).

Value

List of 2:

1. A data frame with the model output in column 1 and the corresponding explanatory data in subsequent columns.
2. A data frame showing the range of `data` compared to the training data, on a 0-1 scale.

References

Halvorsen, R. (2013) A strict maximum likelihood explanation of MaxEnt, and some implications for distribution modelling. Sommerfeltia, 36, 1-132.

Halvorsen, R., Mazzoni, S., Bryn, A. & Bakkestuen, V. (2015) Opportunities for improved distribution modelling practice via a strict maximum likelihood interpretation of MaxEnt. Ecography, 38, 172-183.

Phillips, S.J., Anderson, R.P. & Schapire, R.E. (2006) Maximum entropy modeling of species geographic distributions. Ecological Modelling, 190, 231-259.

Examples

```
## Not run:
modeloutput <- projectModel(newdat,
   transformation = "D:/path/to/modeling/directory/deriveVars/transformations.Rdata",
   model = "D:/path/to/modeling/directory/selectEV/round/model/1.lambdas")

## End(Not run)

proj <- projectModel(toydata_sp1po, toydata_dvs$transformations,
   system.file("extdata/sommerfeltia", "1.lambdas", package = "MIAmaxent"))
proj
```

```
## Not run:
# From vignette:
grasslandPrediction <- projectModel(grasslandPO,
   transformation = grasslandDVs[[2]],
   model = system.file("extdata", "1.lambdas", package = "MIAmaxent"))
head(grasslandPrediction$output)
grasslandPrediction$ranges

# From vignette:
library(raster)
contfiles <- list.files(system.file("extdata", "EV_continuous", package = "MIAmaxent"),
   full.names = TRUE)
catfiles <- list.files(system.file("extdata", "EV_categorical", package = "MIAmaxent"),
   full.names = TRUE)
stack <- raster::stack(c(contfiles, catfiles))
stackpts <- rasterToPoints(stack)
spatialPrediction <- projectModel(stackpts,
   transformation = grasslandDVs[[2]],
   model = system.file("extdata", "1.lambdas", package = "MIAmaxent"))
Predictionraster <- raster(stack, layer=0)
Predictionraster <- rasterize(spatialPrediction$output[, c("x", "y")], Predictionraster,
   field = spatialPrediction$output$PRO)
plot(Predictionraster, colNA="black")

## End(Not run)
```

`readData` *Read in data object from files.*

Description

`readData` reads in occurrence data in CSV file format and environmental data in ASCII raster file format and produces a data object which can be used as the starting point for the functions in this package. This function is intended to make reading in data easy for users familiar with the maxent.jar program. It is emphasized that important considerations for data preparation (e.g. cleaning, sampling bias removal, etc.) are not treated in this package and must be dealt with separately!

Usage

```
readData(occurrence, contEV = NULL, catEV = NULL, maxbkg = 10000,
  PA = FALSE, XY = FALSE)
```

Arguments

`occurrence` Full pathway of the '.csv' file of occurrence data. The first column of the CSV should code occurrence (see Details), while the second and third columns should contain X and Y coordinates corresponding to the ASCII raster coordinate system. The first row is read as a header row.

contEV
: Pathway to a directory containing continuous environmental variables in '.asc' file format.

catEV
: Pathway to a directory containing categorical environmental variables in '.asc' file format.

maxbkg
: Integer. Maximum number of grid cells randomly selected as unknown background points for the response variable. Default is 10,000. Irrelevant for presence/absence data (PA = TRUE) and ignored for presence-only data (PA = FALSE) if occurrence contains 'NA' values.

PA
: Logical. Does occurrence represent presence/absence data? This argument affects how the values in occurrence are interpreted, and controls what type of data object is produced. See details.

XY
: Logical. Include XY coordinates in the output. May be useful for spatial plotting. Note that coordinates included in the training data used to build the model will be treated as explanatory variables.

Details

When occurrence represents presence-only data (PA = FALSE), all rows with values other than 'NA' in column 1 of the CSV file are treated as presence locations. If column 1 contains any values of 'NA', these rows are treated as the unknown background locations for the response variable. Thus, 'NA' can be used to specify a specific set of background locations if desired. Otherwise background points are randomly selected from the full extent of the raster cells which are not already included as presence locations. Only cells which contain data for all environmental variables are selected as background locations.

When occurrence represents presence/absence data (PA = TRUE), rows with value '0' in column 1 of the CSV are treated as absence locations, rows with value 'NA' are excluded, and all other rows are treated as presences.

The names of the ASCII raster files are used as the names of the explanatory variables, so these files should be uniquely named, and the names must not contain spaces, underscores, or colons. Underscores and colons are reserved to denote derived variables and interaction terms repectively. readData automatically replaces underscores with hyphens.

Value

Data frame with the Response Variable (RV) in the first column, and Explanatory Variables (EVs) in subsequent columns. When PA = FALSE, RV values are 1/NA, and when PA = TRUE, RV values are 1/0.

With presence-only occurrence data, the returned output can be used as the data argument for plotFOP, deriveVars, selectDVforEV, selectEV, and plotResp. With presence/absence occurrence data, the returned output can be used as the data argument for testAUC. Output from readData can also be used as the data argument for plotResp2, and projectModel, but for these functions the values of RV are irrelevant.

Examples

```
## Not run:
dat <- readData(occurrence = "D:/path/to/occurrence/data.csv",
```

```
   contEV = "D:/path/to/continuousEV/directory",
   catEV = "D:/path/to/categoricalEV/directory", maxbkg = 100000, XY = TRUE)

## End(Not run)

toydata_sp1po <- readData(system.file("extdata/sommerfeltia", "Sp1.csv", package = "MIAmaxent"),
   contEV = system.file("extdata/sommerfeltia", "EV_continuous", package = "MIAmaxent"))
toydata_sp1po

## Not run:
# From vignette:
grasslandPO <- readData(
   occurrence = system.file("extdata", "occurrence_PO.csv", package = "MIAmaxent"),
   contEV = system.file("extdata", "EV_continuous", package = "MIAmaxent"),
   catEV = system.file("extdata", "EV_categorical", package = "MIAmaxent"),
   maxbkg = 20000)
head(grasslandPO)

# From vignette:
grasslandPA <- readData(
   occurrence = system.file("extdata", "occurrence_PA.csv", package = "MIAmaxent"),
   contEV = system.file("extdata", "EV_continuous", package = "MIAmaxent"),
   catEV = system.file("extdata", "EV_categorical", package = "MIAmaxent"),
   PA = TRUE, XY = TRUE)
head(grasslandPA)
tail(grasslandPA)

## End(Not run)
```

`selectDVforEV` *Select parsimonious sets of derived variables.*

Description

For each explanatory variable (EV), `selectDVforEV` selects the parsimonious set of derived variables (DV) which best explains variation in a given response variable. The function uses a process of forward selection based on comparison of nested models by the F-test. A DV is selected for inclusion when, during nested model comparison, it accounts for a significant amount of remaining variation, under the alpha value specified by the user.

Usage

```
selectDVforEV(data, dvdata, alpha = 0.01, dir = NULL, trainmax = NULL)
```

Arguments

`data`	Data frame containing the response variable in the first column and explanatory variables in subsequent columns. The response variable should represent presence/background data, coded as: 1/NA. See `readData`.

dvdata	List of data frames, with each data frame containing derived variables for a given explanatory variable (e.g. the first item in the list returned by deriveVars).
alpha	Alpha-level used in F-test comparison of models. Default is 0.01.
dir	Directory to which files will be written during subset selection of derived variables. Defaults to the working directory.
trainmax	Integer. Maximum number of uninformed background points to be used to train the models. May be used to reduce computation time for data sets with very large numbers of points. Default is no maximum. See Details for more information.

Details

The F-statistic that `selectDVforEV` uses for nested model comparison is calculated using equation 59 in Halvorsen (2013). See Halvorsen et al. (2015) for a more detailed explanation of the forward selection procedure.

If the derived variables were created using `deriveVars`, the same response variable should be used in `selectDVforEV`, as the deviation and spline transformations produced by `deriveVars` are RV-specific.

If `trainmax` reduces the number of uninformed background points in the training data, a new `data` object is returned as part of the function output. This `data` object shows which of the uninformed background points were randomly selected, and should be used together with the selected DVs in `selectEV` during continued model selection.

Explanatory variables should be uniquely named, and the names must not contain spaces, underscores, or colons. Underscores and colons are reserved to denote derived variables and interaction terms repectively.

Value

List of 2 (3):

1. A list of data frames, with each data frame containing *selected* DVs for a given EV. This item is recommended as input for `dvdata` in `selectEV`.
2. A list of data frames, where each data frame shows the trail of forward selection of DVs for a given EV.
3. (If `trainmax` reduces the number of uninformed background points) a new `data` object. See details.

References

Halvorsen, R. (2013). A strict maximum likelihood explanation of MaxEnt, and some implications for distribution modelling. Sommerfeltia, 36, 1-132.

Halvorsen, R., Mazzoni, S., Bryn, A., & Bakkestuen, V. (2015). Opportunities for improved distribution modelling practice via a strict maximum likelihood interpretation of MaxEnt. Ecography, 38(2), 172-183.

Examples

```
## Not run:
selectedvs <- selectDVforEV(dat, deriveddat, alpha = 0.0001,
   dir = "D:/path/to/modeling/directory")

# From vignette:
grasslandDVselect <- selectDVforEV(grasslandPO, grasslandDVs[[1]], alpha = 0.001)
summary(grasslandDVs$EVDV)
sum(sapply(grasslandDVs$EVDV, length))
summary(grasslandDVselect$selectedDV)
sum(sapply(grasslandDVselect$selectedDV, length))

## End(Not run)
```

---

selectEV — *Select parsimonious set of explanatory variables.*

Description

`selectEV` selects the parsimonious set of explanatory variables (EVs) which best explains variation in a given response variable (RV). Each EV can be represented by 1 or more derived variables (see `deriveVars`). The function uses a process of forward selection based on comparison of nested models by the F-test. An EV is selected for inclusion when, during nested model comparison, it accounts for a significant amount of remaining variation, under the alpha value specified by the user.

Usage

```
selectEV(data, dvdata, alpha = 0.01, interaction = FALSE, dir = NULL,
  trainmax = NULL)
```

Arguments

`data`	Data frame containing the response variable in the first column and explanatory variables in subsequent columns. The response variable should represent presence/background data, coded as: 1/NA. See `readData`.
`dvdata`	List of data frames, with each data frame containing *selected* derived variables for a given explanatory variable (e.g. the first item in the list returned by `selectDVforEV`).
`alpha`	Alpha-level used in F-test comparison of models. Default is 0.01.
`interaction`	Logical. Allows interaction terms between pairs of EVs. Default is `FALSE`.
`dir`	Directory to which files will be written during subset selection of explanatory variables. Defaults to the working directory.
`trainmax`	Integer. Maximum number of uninformed background points to be used to train the models. May be used to reduce computation time for data sets with very large numbers of points. Default is no maximum. See Details for more information.

Details

The F-statistic that `selectEV` uses for nested model comparison is calculated using equation 59 in Halvorsen (2013). See Halvorsen et al. (2015) for a more detailed explanation of the forward selection procedure.

When `interaction = TRUE`, the forward selection procedure selects a parsimonious group of individual EVs first, and then tests interactions between EVs included in the model afterwards. Therefore, interactions are only explored between terms which are individually explain a significant amount of variation. When `interaction = FALSE`, interactions are not considered.

If `trainmax` reduces the number of uninformed background points in the training data, a new `data` object is returned as part of the function output. This `data` object shows which of the uninformed background points were randomly selected, and should be used together with the selected EVs in `plotResp` if plotting single-effect model response.

Explanatory variables should be uniquely named, and the names must not contain spaces, underscores, or colons. Underscores and colons are reserved to denote derived variables and interaction terms repectively.

Value

List of 2 (3):

1. A list of data frames, with one data frame for each *selected* EV. This item is recommended as input for `dvdata` in `plotResp`.
2. A data frame showing the trail of forward selection of individual EVs (and interaction terms if necessary).
3. (If `trainmax` reduces the number of uninformed background points) a new `data` object. See details.

References

Halvorsen, R. (2013). A strict maximum likelihood explanation of MaxEnt, and some implications for distribution modelling. Sommerfeltia, 36, 1-132.

Halvorsen, R., Mazzoni, S., Bryn, A., & Bakkestuen, V. (2015). Opportunities for improved distribution modelling practice via a strict maximum likelihood interpretation of MaxEnt. Ecography, 38(2), 172-183.

Examples

```
## Not run:
selectedevs <- selectEV(dat, selectedderiveddat, alpha = 0.0001,
   dir = "D:/path/to/modeling/directory", interaction = TRUE)

# From vignette:
grasslandEVselect <- selectEV(grasslandPO, grasslandDVselect[[1]], alpha = 0.001,
   interaction = TRUE)
summary(grasslandDVselect[[1]])
length(grasslandDVselect[[1]])
summary(grasslandEVselect[[1]])
length(grasslandEVselect[[1]])
```

```
plot(grasslandEVselect$selection$round, grasslandEVselect$selection$addedFVA)

## End(Not run)
```

---

testAUC *Calculate model AUC with test data.*

Description

For a given model, `testAUC` calculates the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) as a threshold-independent measure of binary classification performance. This function is intended to be used with occurence data that is independent from the data used to train the model, to obtain an unbiased measure of model performance.

Usage

```
testAUC(data, transformation, model)
```

Arguments

`data`	Data frame containing test occurrence data in the first column and corresponding explanatory variables in subsequent columns. The test data should be coded as: 1/0/NA, representing presence, absence, and unknown. See `readData`.
`transformation`	Full pathway of the 'transformation.Rdata' file containing the transformations used to build the model. This file is saved as a result of the `deriveVars` function. Equivalently, the second item in the list returned by `deriveVars` can be used directly.
`model`	Full pathway of the '.lambdas' file of the model in question. This file is saved as a result of `selectEV`.

Value

In addition to returning the testAUC value, graphical output showing the corresponding ROC plot is produced. The point along the ROC curve where the discrimination threshold is PRO = 1 is shown for reference.

Examples

```
## Not run:
AUC <- testAUC(testdat,
   transformation = "D:/path/to/modeling/directory/deriveVars/transformations.Rdata",
   model = "D:/path/to/modeling/directory/selectEV/round/model/1.lambdas")

## End(Not run)

sp1pa <- toydata_sp1po
sp1pa$RV[is.na(sp1pa$RV)] <- 0
```

```
sp1pa[, 1] <- sample(sp1pa$RV, 40)
auc <- testAUC(sp1pa, toydata_dvs$transformations,
   system.file("extdata/sommerfeltia", "1.lambdas", package = "MIAmaxent"))
auc

## Not run:
From vignette:
grasslandAUC <- testAUC(grasslandPA, transformation = grasslandDVs[[2]],
   model = system.file("extdata", "1.lambdas", package = "MIAmaxent"))
grasslandAUC

## End(Not run)
```

`toydata_dvs` *Derived variables and transformation functions, from toy data.*

Description

Derived variables and transformation functions for distribution modeling of a small, synthetic data set used in Halvorsen (2013).

Usage

```
toydata_dvs
```

Format

List with 2 elements:

1. A list of 4 data frames, with each containing the derived variables produced for a given explanatory variable.
2. A list of all the transformation functions used to produce the derived variables.

Source

Produced from `toydata_sp1po` using `deriveVars`.

References

Halvorsen, R. (2013) A strict maximum likelihood explanation of MaxEnt, and some implications for distribution modelling. Sommerfeltia, 36, 1-132.

`toydata_seldvs` *Selected derived variables accompanied by selection trails, from toy data.*

Description

Selected derived variables and tables showing forward model selection of derived variables for distribution modeling of a small, synthetic data set used in Halvorsen (2013).

Usage

```
toydata_seldvs
```

Format

List with 2 elements:

1. A list of 3 data frames, with each containing the derived variables selected for a given explanatory variable.
2. A list of forward model selection trails used to select derived variables.

Source

Produced from `toydata_dvs` and `toydata_sp1po` using `selectDVforEV`.

References

Halvorsen, R. (2013) A strict maximum likelihood explanation of MaxEnt, and some implications for distribution modelling. Sommerfeltia, 36, 1-132.

`toydata_selevs` *Selected explanatory variables accompanied by selection trails, from toy data.*

Description

Selected explanatory variables and tables showing forward model selection of explanatory variables for distribution modeling of a small, synthetic data set used in Halvorsen (2013). Each individual explanatory variable is represented by a group of derived variables.

Usage

```
toydata_selevs
```

Format

List with 2 elements:

1. A list of 5 data frames, where the first 3 represent selected explanatory variables, and the last 2 represent selected interaction terms between these explanatory variables.
2. A trail of forward model selection used to select explanatory variables and interaction terms.

Source

Produced from `toydata_seldvs` and `toydata_sp1po` using `selectEV`.

References

Halvorsen, R. (2013) A strict maximum likelihood explanation of MaxEnt, and some implications for distribution modelling. Sommerfeltia, 36, 1-132.

`toydata_sp1po` *Occurrence and environmental toy data.*

Description

A small, synthetic data set for distribution modeling, consisting of occurrence and environmental data, from Halvorsen (2013). The study area consists of 40 grid cells, with 8 row and 5 columns, in which 10 presences occur.

Usage

```
toydata_sp1po
```

Format

A data frame with 40 rows and 5 variables:

RV response variable, occurrence either presence or uninformed background

EV11 explanatory variable: northing

EV12 explanatory variable: easting

EV13 explanatory variable: modified random uniform

EV14 explanatory variable: random uniform

Source

Halvorsen, R. (2013) A strict maximum likelihood explanation of MaxEnt, and some implications for distribution modelling. Sommerfeltia, 36, 1-132.

Index