

Package ‘ecotrends’

December 9, 2025

Type Package

Title Temporal Trends in Ecological Niche Models

Version 1.2

Maintainer A. Marcia Barbosa <ana.marcia.barbosa@gmail.com>

Description Computes temporal trends in environmental suitability obtained from ecological niche models, based on a set of species presence point coordinates and predictor variables.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

URL <https://github.com/AMBarbosa/ecotrends>

BugReports <https://github.com/AMBarbosa/ecotrends/issues>

Depends R (>= 3.6)

Config/testthat/edition 3

Imports fuzzySim (>= 4.26), maxnet, modEvA (>= 3.21), terra (>= 1.4-19), trend

NeedsCompilation no

Author A. Marcia Barbosa [aut, cre],
João Alírio [aut],
Nuno Garcia [aut],
João Campos [aut],
Ana Cláudia Teodoro [aut],
Lia Bárbara Duarte [aut],
Isabel Pôças [aut],
Salvador Arenas-Castro [aut],
Neftalí Sillero [aut]

Repository CRAN

Date/Publication 2025-12-09 14:00:02 UTC

Contents

getImportance	2
getModels	3
getPerformance	6
getPredictions	7
getTrend	8
getVariables	10
pixelArea	11
varsAvailable	13
Index	14

getImportance	<i>Get variable importance</i>
---------------	--------------------------------

Description

This function computes the permutation importance of each variable in each model, by shuffling each variable in turn (a given number of times) and computing the root mean squared difference between the actual model predictions and those obtained with the shuffled variable. Values are then normalized to a percentage by dividing each by the sum of all values and multiplying by 100. Note that "importance" is not a straightforward concept and it can be measured in many different ways. Permutation importance is a widely used and effective model-agnostic technique, which provides an intuitive measure of the contribution of each variable, particularly in complex and non-linear models.

Usage

```
getImportance(  
  mods,  
  nper = 10,  
  verbosity = 2,  
  plot = TRUE,  
  palette = "Dark2",  
  ...  
)
```

Arguments

mods	output of <code>getModels()</code> .
nper	integer value (default 10; increase for more accurate but computationally intensive results) indicating the number of permutations for shuffling each variable.
verbosity	integer value indicating the amount of messages to display. The default is 2, for the maximum number of messages available.

plot	logical value specifying whether to produce a line (spaghetti) plot of the mean importance of each variable along the periods. Note that this plot does not reflect the deviations around this mean, and that it may become hard to read if there are many variables or if their importances overlap.
palette	argument to pass to <code>hcl.colors()</code> specifying the colours for the lines (if <code>plot=TRUE</code>). The default is "Dark2"; run <code>hcl.pals()</code> for other options.
...	additional arguments that can be passed to <code>base::plot()</code> , e.g. 'main', 'xlab', 'ylab' or 'las'.

Value

A data frame with the permutation importance (expressed as percentage) of each variable in each model replicate for each period, along with the cross-replicate mean and standard deviation. If `plot=TRUE` (the default), also a spaghetti plot of mean variable importance per period.

Author(s)

A. Marcia Barbosa

See Also

`varImportance` in package **predicts**; `bm_VariablesImportance` in package **biomod2**

Examples

```
# Several data prep steps required.
# See https://github.com/AMBarbosa/ecotrends for a full worked example.
```

getModels

Get models

Description

This function computes `maxnet::maxnet()` ecological niche models for a set of time steps or periods (e.g. years), given environmental layers for each period and a set of presence point coordinates.

Usage

```
getModels(
  occs,
  rasts,
  region = NULL,
  nbg = 10000,
  seed = NULL,
  bias = FALSE,
  collin = TRUE,
  maxcor = 0.75,
```

```

    maxvif = 5,
    classes = "default",
    regmult = 1,
    nreps = 10,
    test = 0.2,
    file = NULL,
    verbosity = 2
)

```

Arguments

<code>occs</code>	species occurrence coordinates (2 columns in this order: x, y or LONGitude, LATitude) in an object coercible to a data.frame (e.g. a data.frame, matrix, tibble, sf object or SpatVector of points), and in the same coordinate reference system as 'rasts'
<code>rasts</code>	(multi-layer) SpatRaster with a time series of variables to use in the models. The layers should be ordered chronologically, and their names should be in the form "varname [underscore] time", e.g. "tmin_1985" or "tmin_1" (with no more underscores than this), as in the output of <code>getVariables()</code> . Note that, if a variable has no spatial variation in a given time step or period, it is excluded (as it cannot have an effect) from that model, so in practice not all models will include the exact same set of variables. If verbosity > 1, messages will report which variables were excluded from each period.
<code>region</code>	optional SpatExtent or SpatVector polygon delimiting the region of 'rasts' within which to compute the models. The default is NULL, to use the entire extent of 'rasts' with pixel values. Note that 'region' should ideally include only reasonably surveyed areas that are accessible to the species, as pixels that don't overlap presence points are taken by Maxent as available and unoccupied.
<code>nbg</code>	integer value indicating the maximum number of background pixels to select randomly for use in the models. The default is 10,000, or the total number of non-NA pixels in 'rasts' if that's less.
<code>seed</code>	optional integer value to pass to <code>set.seed()</code> specifying the random seed to use for sampling the background pixels (if 'nbg' is smaller than the number of pixels in 'rasts') and for extracting the test samples (if nreps > 0).
<code>bias</code>	argument to pass to <code>fuzzySim::selectAbsences()</code> specifying if/how the selection of unoccupied background points should be biased to incorporate survey effort. Can be TRUE to make selection more likely towards the vicinity of occurrence points (which may indicate that those areas have been surveyed); or a SpatRaster of weights (bias layer), with the same coordinate reference system as 'occs' and 'rasts', with higher values where selection should be proportionally more likely, and zero or NA where points should not be placed. Default FALSE.
<code>collin</code>	logical value indicating whether multicollinearity among the variables should be reduced prior to computing each model. The default is TRUE, in which case the <code>fuzzySim::corSelect()</code> and <code>fuzzySim::multicol()</code> functions are used (<code>collinear::collinear()</code> was used previously, but we eliminated the dependency on this package after it introduced breaking changes for the 2nd time). Note that, if the collinearity structure varies among periods, the set of included variables may

	also vary. If verbosity > 1, messages will report which variables were excluded from each period.
maxcor	numeric value to pass to <code>fuzzySim::corSelect()</code> (if <code>collin = TRUE</code> , and using <code>select = "cor"</code>) indicating the maximum correlation allowed between any pair of predictor variables. The default is 0.75.
maxvif	numeric value (used if <code>collin = TRUE</code>) indicating the maximum VIF allowed for selected predictor variables. VIF values are computed with <code>fuzzySim::multicol()</code> and variables are excluded in a stepwise manner, starting from the one with the highest VIF and recomputing every time, until none exceeds the 'maxvif' value. The default is 5.
classes	character value to pass to <code>maxnet::maxnet.formula()</code> indicating the continuous feature classes desired. Can be "default" or any subset of "lqpht" (linear, quadratic, product, hinge, threshold) – for example, "lqh" for just linear, quadratic and hinge features. See References for guidance.
regmult	numeric value to pass to <code>maxnet::maxnet()</code> indicating the constant to adjust regularization. The default is 1. See References for guidance.
nreps	integer value indicating the number of train-test datasets for testing the models. The default is 10. With <code>nreps = 0</code> , there is no division of the dataset into train and test samples, so models are trained on the entire dataset for each period. If <code>nreps > 0</code> , presences are randomly assigned to the train and test sample in each replicate (in the proportion defined by the 'test' argument), while the background remains the same.
test	(if <code>nreps > 0</code>) numeric value indicating the proportion of presences to set aside for testing each model. The default is 0.2, i.e. 20%.
file	optional file name (including path, not including extension) if you want the output list of model objects to be saved on disk. If 'file' already exists in the working directory (meaning that models were already computed), models are imported from there.
verbosity	integer value indicating the amount of messages to display. The default is 2, for the maximum number of messages available.

Value

A list of three elements:

\$models: a list of lists of model objects of class `maxnet::maxnet`. Each element of the list corresponds to a period (e.g. period), and each sub-element a replicate.

\$data: a data frame with the presences, remaining background points and their environmental values used in the model(s).

Author(s)

A. Marcia Barbosa

References

Elith J., Phillips S.J., Hastie T., Dudik M., Chee Y.E., Yates, C.J. (2011) A Statistical Explanation of MaxEnt for Ecologists. *Diversity and Distributions* 17:43-57. <http://dx.doi.org/10.1111/j.1472-4642.2010.00725.x>

Merow C., Smith M.J., Silander J.A. (2013) A practical guide to MaxEnt for modeling species' distributions: what it does, and why inputs and settings matter. *Ecography* 36:1058-1069. <https://doi.org/10.1111/j.1600-0587.2013.07872.x>

See Also

`maxnet::maxnet()`

Examples

```
# Several data prep steps required.
# See https://github.com/AMBarbosa/ecotrends for a full worked example.
```

getPerformance	<i>Get model performance</i>
----------------	------------------------------

Description

Get model performance

Usage

```
getPerformance(
  rasts,
  mods,
  metrics = c("AUC", "TSS", "kappa"),
  plot = FALSE,
  verbosity = 2
)
```

Arguments

<code>rasts</code>	SpatRasterDataset output of <code>getPredictions()</code> , or 'file' argument previously provided to <code>getPredictions()</code>
<code>mods</code>	output of <code>getModels()</code>
<code>metrics</code>	character vector with the metrics to compute. Can be any subset (default all) of <code>c("AUC", "TSS", "kappa")</code> , with the latter two computed at their maximum values (optimal thresholds). Performance metrics are computed with presence against all background (using 'modEvA' package functions with <code>pbg=TRUE</code>), so they evaluate the capacity of distinguishing presence from random, rather than presence from absence pixels (Phillips et al., 2006).

plot	logical value indicating whether plots should also be produced to illustrate the performance metrics for each model. The default is FALSE; TRUE can be slow for large datasets.
verbosity	integer value indicating the amount of messages to display. The default is 2, for the maximum number of messages available.

Value

This function returns a data frame of the performance evaluation results for each model.

References

Phillips, S.J., Anderson, R.P., Schapire, R.E. (2006) Maximum entropy modeling of species geographic distributions. *Ecological Modelling*, 190: 231-259. <https://doi.org/10.1016/j.ecolmodel.2005.03.026>

Examples

```
# Several data prep steps required.
# See https://github.com/AMBarbosa/ecotrends for a full worked example.
```

getPredictions	<i>Get predictions</i>
----------------	------------------------

Description

Get predictions

Usage

```
getPredictions(
  rasts,
  mods,
  region = NULL,
  type = "cloglog",
  clamp = TRUE,
  file = NULL,
  verbosity = 2
)
```

Arguments

rasts	(multi-layer) SpatRaster containing the variables (with the same names) used in the models.
mods	output of <code>getModels()</code> .
region	optional SpatExtent or SpatVector polygon delimiting the region of 'rasts' within which to compute the predictions. The default is NULL, to use the entire extent of 'rasts'. Note that predictions may be unreliable outside the 'region' used for <code>getModels()</code> , as they are extrapolating beyond the analysed conditions.

type	character value to pass to <code>predict()</code> indicating the type of response to compute. Can be "cloglog" (the default and currently recommended), "logistic" (previously but no longer recommended) (Phillips et al., 2017), "exponential", or "link".
clamp	logical value to pass to <code>predict()</code> indicating whether predictors and features should be restricted to the range seen during model training. Default TRUE.
file	optional folder name (including path within the working directory) if you want the prediction rasters to be saved to disk. If 'file' already exists in the working directory (meaning that predictions were already computed), predictions are imported from there.
verbosity	integer value indicating the amount of messages to display. The default is 2, for the maximum number of messages available.

Value

This function returns a `SpatRasterDataset` with one sub-dataset per period, each of which is a (multi-layer) `SpatRaster` with the predictions of each replicate (if there are replicates) for that period.

Author(s)

A. Marcia Barbosa

References

Phillips, S.J., Anderson, R.P., Dudik, M., Schapire, R.E., Blair, M.E., 2017. Opening the black box: an open-source release of Maxent. *Ecography* 40, 887-893. <https://doi.org/10.1111/ecog.03049>

Examples

```
# Several data prep steps required.
# See https://github.com/AMBarbosa/ecotrends for a full worked example.
```

getTrend

Get trend

Description

This function uses `terra::app()` to apply the `trend::sens.slope()` function to each pixel of a multi-layer time series `SpatRaster`, testing for a monotonic (either increasing or decreasing) linear trend in the raster values, as well as the confidence interval of the slope.

Usage

```
getTrend(
  rasts,
  occs = NULL,
  alpha = 0.05,
  conf.level = 0.95,
  file = NULL,
  full = TRUE,
  verbosity = 2
)
```

Arguments

<code>rasts</code>	multi-layer <code>SpatRaster</code> with the output of <code>getPredictions()</code> , or another time series of values for which to detect a trend. Note that >3 non-NA values (i.e. more than 3 time steps) are required for a trend to be computed. If there are >1 replicates per year, their pixel-wise mean is computed prior to analysing the trend.
<code>occs</code>	<code>SpatVector</code> of species occurrence points, or their spatial coordinates (2 columns in this order: x, y or LONGitude, LATitude) in an object coercible to a <code>data.frame</code> (e.g. a <code>data.frame</code> , <code>matrix</code> , <code>tibble</code> , <code>sf</code> object), and in the same coordinate reference system as 'rasts'. If provided, output pixels that do not overlap these points will be NA
<code>alpha</code>	numeric value indicating the threshold significance level for Sen's slope. Default 0.05. Pixels with p-value above this will have NA value in the output.
<code>conf.level</code>	numeric value to pass to <code>trend::sens.slope()</code> indicating the confidence level for the slope of the trend. Default 0.95.
<code>file</code>	optional file name (including path, not including extension) if you want the outputs raster(s) to be imported from or saved to disk.
<code>full</code>	logical value indicating whether to output a multi-layer raster with the full results of the Mann-Kendall test (namely the Tau value, p-value, S, and variance of S) and the results of Sen's slope calculation (slope estimate, p-value, upper and lower confidence limit). If set to FALSE, a single-layer raster will be returned with the (significant) Sen's slope values.
<code>verbosity</code>	integer value indicating the amount of messages to display. The default is 2, for the maximum number of messages available.

Value

If `full=FALSE`, a single-layer `SpatRaster` where each pixel (or each pixel with occurrence points, if 'occs' is not NULL) has the value of Sen's slope (positive if increasing, negative if decreasing), or NA if the trend is non-significant (i.e., if the p-value is larger than the specified 'alpha'). If `full=TRUE` (the default), additional layers are produced with associated statistics, including the lower and upper bounds of Sen's slope (given the input 'conf.level').

Author(s)

A. Marcia Barbosa

See Also

`trend::sens.slope()`, `trend::mk.test()`, `Kendall::MannKendall()`, `spatialEco::raster.kendall()`

Examples

```
# Several data prep steps required.
# See https://github.com/AMBarbosa/ecotrends for a full worked example.
```

getVariables

Get variables

Description

This function downloads a specified time series of variables from a specified environmental data source, optionally (to save download time) within a specified region. Currently "TerraClimate" (Abatzoglou et al., 2018) is the only implemented data source.

Usage

```
getVariables(
  source = "TerraClimate",
  vars = varsAvailable(source)$vars,
  years = varsAvailable(source)$years,
  region = c(-180, 180, -90, 90),
  file = NULL,
  verbosity = 2
)
```

Arguments

source	source to import the raster variables from, e.g. "TerraClimate" (currently the only one implemented)
vars	character vector of the names of the variables to be imported. Run <code>varsAvailable()</code> for options. The default is to download all available variables from the specified 'source'. Note that the download can take a long time, especially for many variables, long series of years, and/or large regions.
years	year range to get the variables from (e.g. 1979:2013). Note that the download can take a long time for long series of years.
region	optional length-four numeric vector (xmin, xmax, ymin, ymax geodetic coordinates in degrees), <code>SpatExtent</code> or <code>SpatVector</code> polygon delimiting the region of the world for which the variables should be downloaded. See e.g. <code>fuzzySim::getRegion()</code> for suggestions. The larger the region, the longer the download time.
file	optional file name (including the path, not including the filename extension) if you want the downloaded variable rasters to be saved on disk, in which case they are saved as a compressed multi-layer GeoTIFF. If 'file' already exists in the working directory, variables are imported from there.
verbosity	integer value indicating the amount of messages to display. The default is 2, for the maximum number of messages available.

Value

multi-layer SpatRaster

Author(s)

A. Marcia Barbosa

References

Abatzoglou, J.T., S.Z. Dobrowski, S.A. Parks, K.C. Hegewisch (2018) Terraclimate, a high-resolution global dataset of monthly climate and climatic water balance from 1958-2015. Scientific Data, 5, Article number: 170191. doi: 10.1038/sdata.2017.191(2018). Database URL: <https://www.climatologylab.org/terraclimate.h>

See Also

[varsAvailable\(\)](#)

Examples

```
if (interactive()) {

  # note these downloads may take long!

  vars <- ecotrends::getVariables(vars = c("tmin", "tmax", "ppt", "pet", "ws"),
    years = 1981:1990, region = terra::ext(-11, -4, 37, 45), file = paste0(tempdir(), "/variables"))

  # tempdir() is here to comply with CRAN policy, but you should normally
  # use a directory that you can access again when reopening R
  # to avoid downloading the variables again every time

  names(vars)

  terra::plot(vars[[1:6]])

}
```

pixelArea

Average pixel area

Description

This function uses the [terra::cellSize\(\)](#) function to compute (and optionally map) the area covered by each (optionally non-NA) pixel in a raster map; and then it computes either the mean or the centroid pixel area in that map, for an idea of pixel size in that region. Pixel size can vary widely across latitudes, especially in unprojected longitude-latitude rasters, but also in rasters projected to other non-equal-area coordinate reference systems.

Usage

```
pixelArea(  
  rast,  
  type = "mean",  
  unit = "m",  
  mask = TRUE,  
  map = TRUE,  
  verbosity = 2  
)
```

Arguments

<code>rast</code>	SpatRaster for which to compute the pixel area
<code>type</code>	character value indicating whether the output value should be the "mean" (default) or "centroid" pixel area
<code>unit</code>	numeric value indicating the units for the output value: either "m" (default) or "km" squared.
<code>mask</code>	logical value (default TRUE) indicating whether to consider only the areas of non-NA pixels
<code>map</code>	logical value (default TRUE) indicating whether to also plot a map
<code>verbosity</code>	integer value indicating the amount of messages to display. The default is 2, for the maximum number of messages available.

Value

numeric value

Author(s)

A. Marcia Barbosa, wrapping 'terra' functions by Robert H. Hijmans

See Also

[terra::cellSize\(\)](#), which this function wraps

Examples

```
r <- terra::rast(system.file("ex/elev.tif", package = "terra"))  
  
pixelArea(r)  
  
pixelArea(r, unit = "km")
```

varsAvailable	<i>Variables available</i>
---------------	----------------------------

Description

Variables available

Usage

```
varsAvailable(source = NULL)
```

Arguments

source	the source database where to check available variables. By default (i.e. if left NULL), all implemented sources are used. Currently "TerraClimate" (https://www.climatologylab.org/terraclimate.html) is the only implemented source.
--------	--

Value

a named nested list with one element per data source, each of which containing two elements: a character vector of the names of the variables available from that source, and an integer vector of the years for which those variables are available.

Author(s)

A. Marcia Barbosa

References

Abatzoglou, J.T., S.Z. Dobrowski, S.A. Parks, K.C. Hegewisch (2018) Terraclimate, a high-resolution global dataset of monthly climate and climatic water balance from 1958-2015. Scientific Data, 5, Article number: 170191. doi: 10.1038/sdata.2017.191(2018). Database URL: <https://www.climatologylab.org/terraclimate.html>

See Also

[getVariables\(\)](#)

Examples

```
varsAvailable()
```

Index

`base::plot()`, 3

`fuzzySim::corSelect()`, 4, 5
`fuzzySim::multicol()`, 4, 5
`fuzzySim::selectAbsences()`, 4

`getImportance`, 2
`getModels`, 3
`getModels()`, 2, 6, 7
`getPerformance`, 6
`getPredictions`, 7
`getPredictions()`, 6, 9
`getTrend`, 8
`getVariables`, 10
`getVariables()`, 4, 13

`hcl.colors()`, 3

`maxnet::maxnet`, 5
`maxnet::maxnet()`, 3, 5, 6
`maxnet::maxnet.formula()`, 5

`pixelArea`, 11
`predict()`, 8

`set.seed()`, 4

`terra::app()`, 8
`terra::cellSize()`, 11, 12
`trend::mk.test()`, 10
`trend::sens.slope()`, 8–10

`varsAvailable`, 13
`varsAvailable()`, 11