

Package: glossa (via r-universe)

October 16, 2024

Type Package

Title User-Friendly 'shiny' App for Bayesian Species Distribution Models

Version 1.0.0

Description A user-friendly 'shiny' application for Bayesian machine learning analysis of marine species distributions. GLOSSA (Global Species Spatiotemporal Analysis) uses Bayesian Additive Regression Trees (BART; Chipman, George, and McCulloch (2010) <[doi:10.1214/09-AOAS285](https://doi.org/10.1214/09-AOAS285)>) to model species distributions with intuitive workflows for data upload, processing, model fitting, and result visualization. It supports presence-absence and presence-only data (with pseudo-absence generation), spatial thinning, cross-validation, and scenario-based projections. GLOSSA is designed to facilitate ecological research by providing easy-to-use tools for analyzing and visualizing marine species distributions across different spatial and temporal scales.

License GPL-3

URL <https://github.com/iMARES-group/glossa>,
<https://iMARES-group.github.io/glossa/>

BugReports <https://github.com/iMARES-group/glossa/issues>

Depends bs4Dash, R (>= 4.0.0), shiny

Imports dbarts, dplyr, DT, GeoThinner, ggplot2, htmltools, jsonlite, leaflet, markdown, mcp, pROC, sf, shinyWidgets, sparkline, svglite, terra, tidyterra, waiter, zip

Suggests knitr, matrixStats, rmarkdown, testthat (>= 3.0.0), tidyr, tidyverse

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.3.1

NeedsCompilation no

Author Jorge Mestre-Tomás [aut, cre]

(<<https://orcid.org/0000-0002-8983-3417>>), Alba Fuster-Alonso

[aut] (<<https://orcid.org/0000-0002-7283-291X>>)

Maintainer Jorge Mestre-Tomás <jorge.mestre.tomas@csic.es>

Repository CRAN

Date/Publication 2024-10-15 15:10:06 UTC

Contents

buffer_polygon	2
clean_coordinates	3
create_coords_layer	4
cv_bart	4
extract_noNA_cov_values	5
fit_bart_model	6
generate_pseudo_absences	6
glossa_analysis	7
invert_polygon	8
layer_mask	9
pa_optimal_cutoff	9
predict_bart	10
remove_duplicate_points	10
remove_points_polygon	11
response_curve_bart	11
run_glossa	12
variable_importance	13

Index **14**

buffer_polygon	<i>Enlarge/Buffer a Polygon</i>
----------------	---------------------------------

Description

This function enlarges a polygon by applying a buffer.

Usage

```
buffer_polygon(polygon, buffer_distance)
```

Arguments

polygon	An sf object representing the polygon to be buffered.
buffer_distance	Numeric. The buffer distance in decimal degrees (arc degrees).

Value

An sf object representing the buffered polygon.

clean_coordinates	<i>Clean Coordinates of Presence/Absence Data</i>
-------------------	---

Description

This function cleans coordinates of presence/absence data by removing NA coordinates, rounding coordinates if specified, removing duplicated points, and removing points outside specified spatial polygon boundaries.

Usage

```
clean_coordinates(  
  df,  
  study_area,  
  overlapping = FALSE,  
  decimal_digits = NULL,  
  coords = c("decimalLongitude", "decimalLatitude"),  
  by_timestamp = TRUE,  
  seed = NULL  
)
```

Arguments

df	A dataframe object with rows representing points. Coordinates are in WGS84 (EPSG:4326) coordinate system.
study_area	A spatial polygon in WGS84 (EPSG:4326) representing the boundaries within which coordinates should be kept.
overlapping	Logical indicating whether points overlapping the polygon should be removed (TRUE) or kept (FALSE).
decimal_digits	An integer specifying the number of decimal places to which coordinates should be rounded.
coords	Character vector specifying the column names for longitude and latitude.
by_timestamp	If TRUE, clean coordinates taking into account different time periods defined in the column 'timestamp'.
seed	Optional; an integer seed for reproducibility of results.

Details

This function takes a data frame containing presence/absence data with longitude and latitude coordinates, a spatial polygon representing boundaries within which to keep points, and parameters for rounding coordinates and handling duplicated points. It returns a cleaned data frame with valid coordinates within the specified boundaries.

Value

A cleaned data frame containing presence/absence data with valid coordinates.

create_coords_layer *Create Geographic Coordinate Layers*

Description

Generates raster layers for longitude and latitude from given raster data, applies optional scaling, and restricts the output to a specified spatial mask.

Usage

```
create_coords_layer(layers, study_area = NULL, scale_layers = FALSE)
```

Arguments

layers	Raster or stack of raster layers to derive geographic extent and resolution.
study_area	Spatial object for masking output layers.
scale_layers	Logical indicating if scaling is applied. Default is FALSE.

Value

Raster stack with layers lon and lat.

cv_bart *Cross-Validation for BART Model*

Description

This function performs k-fold cross-validation for a Bayesian Additive Regression Trees (BART) model using presence-absence data and environmental covariate layers. It calculates various performance metrics for model evaluation.

Usage

```
cv_bart(data, k = 10, seed = NULL)
```

Arguments

data	Data frame with a column (named 'pa') indicating presence (1) or absence (0) and columns for the predictor variables.
k	Integer; number of folds for cross-validation (default is 10).
seed	Optional; random seed.

Value

A data frame containing the true positives (TP), false positives (FP), false negatives (FN), true negatives (TN), and various performance metrics including precision (PREC), sensitivity (SEN), specificity (SPC), false discovery rate (FDR), negative predictive value (NPV), false negative rate (FNR), false positive rate (FPR), F-score, accuracy (ACC), balanced accuracy (BA), and true skill statistic (TSS) for each fold.

extract_noNA_cov_values

Extract Non-NA Covariate Values

Description

This function extracts covariate values for species occurrences, excluding NA values.

Usage

```
extract_noNA_cov_values(data, covariate_layers, predictor_variables)
```

Arguments

data A data frame containing species occurrence data with columns x/long (first column) and y/lat (second column).

covariate_layers A list of raster layers representing covariates.

predictor_variables Variables to select from all the layers.

Details

This function extracts covariate values for each species occurrence location from the provided covariate layers. It returns a data frame containing species occurrence data with covariate values, excluding any NA values.

Value

A data frame containing species occurrence data with covariate values, excluding NA values.

fit_bart_model	<i>Fit a BART Model Using Environmental Covariate Layers</i>
----------------	--

Description

This function fits a Bayesian Additive Regression Trees (BART) model using presence/absence data and environmental covariate layers.

Usage

```
fit_bart_model(y, x, seed = NULL)
```

Arguments

y	A numeric vector indicating presence (1) or absence (0).
x	A data frame with the same number of rows as the length of the vector 'y', containing the covariate values.
seed	An optional integer value for setting the random seed for reproducibility.

Value

A BART model object.

generate_pseudo_absences	<i>Generate Pseudo-Absence Points Based on Presence Points, Covariates, and Study Area Polygon</i>
--------------------------	--

Description

This function generates pseudo-absence points within the study area.

Usage

```
generate_pseudo_absences(
  presences,
  study_area,
  raster_stack,
  predictor_variables,
  coords = c("decimalLongitude", "decimalLatitude"),
  decimal_digits = NULL,
  attempts = 100
)
```

Arguments

presences	Data frame containing presence points.
study_area	Spatial polygon defining the study area ('sf' object).
raster_stack	'SpatRaster' object containing covariate data.
predictor_variables	Character vector of the predictor variables selected for this species.
coords	Character vector specifying the column names for latitude and longitude. Defaults to 'c("decimalLongitude", "decimalLatitude")'.
decimal_digits	An integer specifying the number of decimal places to which coordinates should be rounded.
attempts	Integer specifying the number of attempts to generate exact pseudo-absences. Defaults to 100.

Value

Data frame containing both presence and pseudo-absence points.

`glossa_analysis` *Main Analysis Function for GLOSSA Package*

Description

This function wraps all the analysis that the GLOSSA package performs. It processes presence-absence data, environmental covariates, and performs species distribution modeling and projections under past and future scenarios.

Usage

```
glossa_analysis(
  pa_data = NULL,
  fit_layers = NULL,
  proj_files = NULL,
  study_area_poly = NULL,
  predictor_variables = NULL,
  decimal_digits = NULL,
  scale_layers = FALSE,
  buffer = NULL,
  native_range = NULL,
  suitable_habitat = NULL,
  other_analysis = NULL,
  seed = NA,
  waiter = NULL
)
```

Arguments

pa_data	A list of data frames containing presence-absence data.
fit_layers	A SpatRaster stack containing model fitting environmental layers.
proj_files	A list of file paths containing environmental layers for projection scenarios.
study_area_poly	A spatial polygon defining the study area.
predictor_variables	A list of predictor variables to be used in the analysis.
decimal_digits	An integer specifying the number of decimal places to which coordinates should be rounded.
scale_layers	Logical; if TRUE, covariate layers will be scaled based on fit layers.
buffer	Buffer value or distance in decimal degrees (arc_degrees).
native_range	A vector of scenarios ('fit_layers', 'projections') where native range modeling should be performed.
suitable_habitat	A vector of scenarios ('fit_layers', 'projections') where habitat suitability modeling should be performed.
other_analysis	A vector of additional analyses to perform (e.g., 'variable_importance', 'functional_responses', 'cross_validation').
seed	Optional; an integer seed for reproducibility of results.
waiter	Optional; a waiter instance to update progress in a Shiny application.

Value

A list containing structured outputs from each major section of the analysis, including model data, projections, variable importance scores, and habitat suitability assessments.

invert_polygon	<i>Invert a Polygon</i>
----------------	-------------------------

Description

This function inverts a polygon by calculating the difference between the bounding box and the polygon.

Usage

```
invert_polygon(polygon, bbox = NULL)
```

Arguments

polygon	An sf object representing the polygon to be inverted.
bbox	Optional. An sf or bbox object representing the bounding box. If NULL, the bounding box of the input polygon is used.

Value

An sf object representing the inverted polygon.

layer_mask	<i>Apply Polygon Mask to Raster Layers</i>
------------	--

Description

This function crops and extends raster layers to a study area extent (bbox) defined by longitude and latitude then applies a mask based on a provided spatial polygon to remove areas outside the polygon.

Usage

```
layer_mask(layers, study_area)
```

Arguments

layers	A stack of raster layers ('SpatRaster' object) to be processed.
study_area	A spatial polygon ('sf' object) used to mask the raster layers.

Value

A 'SpatRaster' object representing the masked raster layers.

pa_optimal_cutoff	<i>Optimal Cutoff for Presence-Absence Prediction</i>
-------------------	---

Description

This function calculates the optimal cutoff for presence-absence prediction using a BART model.

Usage

```
pa_optimal_cutoff(y, x, model, seed = NULL)
```

Arguments

y	Vector indicating presence (1) or absence (0).
x	Dataframe with same number of rows as the length of the vector 'y' with the covariate values.
model	A BART model object.
seed	Random seed for reproducibility.

Value

The optimal cutoff value for presence-absence prediction.

predict_bart *Make Predictions Using a BART Model*

Description

This function makes predictions using a Bayesian Additive Regression Trees (BART) model on a stack of environmental covariates ('SpatRaster').

Usage

```
predict_bart(bart_model, layers, cutoff = NULL)
```

Arguments

bart_model	A BART model object obtained from fitting BART using the 'dbarts' package.
layers	A SpatRaster object containing environmental covariates for prediction.
cutoff	An optional numeric cutoff value for determining potential presences. If NULL, potential presences and absences will not be computed.

Value

A SpatRaster containing the mean, median, standard deviation, and quantiles of the posterior predictive distribution, as well as a potential presences layer if cutoff is provided.

remove_duplicate_points
Remove Duplicated Points from a Dataframe

Description

This function removes duplicated points from a dataframe based on specified coordinate columns.

Usage

```
remove_duplicate_points(df, coords = c("decimalLongitude", "decimalLatitude"))
```

Arguments

df	A dataframe object with each row representing one point.
coords	A character vector specifying the names of the coordinate columns used for identifying duplicate points. Default is c("decimalLongitude", "decimalLatitude").

Value

A dataframe without duplicated points.

remove_points_polygon *Remove Points Inside or Outside a Polygon*

Description

This function removes points from a dataframe based on their location relative to a specified polygon.

Usage

```
remove_points_polygon(  
  df,  
  polygon,  
  overlapping = FALSE,  
  coords = c("decimalLongitude", "decimalLatitude")  
)
```

Arguments

df	A dataframe object with rows representing points.
polygon	An sf polygon object defining the region for point removal.
overlapping	Logical indicating whether points overlapping the polygon should be removed (TRUE) or kept (FALSE).
coords	Character vector specifying the column names for longitude and latitude. Default is c("decimalLongitude", "decimalLatitude").

Value

A dataframe containing the filtered points.

response_curve_bart *Calculate Response Curve Using BART Model*

Description

This function calculates the response curve (functional responses) using a Bayesian Additive Regression Trees (BART) model.

Usage

```
response_curve_bart(bart_model, data, predictor_names)
```

Arguments

bart_model	A BART model object obtained from fitting BART ('dbarts::bart').
data	A data frame containing the predictor variables (the design matrix) used in the BART model.
predictor_names	A character vector containing the names of the predictor variables.

Value

A list containing a data frame for each independent variable with mean, 2.5th percentile, 97.5th percentile, and corresponding values of the variables.

run_glossa	<i>Run GLOSSA Shiny App</i>
------------	-----------------------------

Description

This function launches the GLOSSA Shiny web application.

Usage

```
run_glossa(
  request_size_mb = 2000,
  launch.browser = TRUE,
  port = getOption("shiny.port")
)
```

Arguments

request_size_mb	Maximum request size for file uploads, in megabytes. Default is 2000 MB.
launch.browser	Logical indicating whether to launch the app in the browser (default is TRUE).
port	Port number for the Shiny app. Uses the port specified by 'getOption("shiny.port")' by default.

Details

The GLOSSA Shiny app provides an interactive interface for users to access GLOSSA functionalities.

Value

No return value, called to launch the GLOSSA app.

Examples

```
if(interactive()) {
  run_glossa()
}
```

variable_importance	<i>Variable Importance in BART Model</i>
---------------------	--

Description

This function computes the variable importance scores for a fitted BART (Bayesian Additive Regression Trees) model using a permutation-based approach. It measures the impact of each predictor variable on the model's performance by permuting the values of that variable and evaluating the change in performance (F-score is the performance metric).

Usage

```
variable_importance(bart_model, cutoff = 0, n_repeats = 10, seed = NULL)
```

Arguments

bart_model	A BART model object.
cutoff	A numeric threshold for converting predicted probabilities into presence-absence.
n_repeats	An integer indicating the number of times to repeat the permutation for each variable.
seed	An optional seed for random number generation.

Value

A data frame where each column corresponds to a predictor variable, and each row contains the variable importance scores across permutations.

Index

buffer_polygon, [2](#)

clean_coordinates, [3](#)
create_coords_layer, [4](#)
cv_bart, [4](#)

extract_noNA_cov_values, [5](#)

fit_bart_model, [6](#)

generate_pseudo_absences, [6](#)
glossa_analysis, [7](#)

invert_polygon, [8](#)

layer_mask, [9](#)

pa_optimal_cutoff, [9](#)
predict_bart, [10](#)

remove_duplicate_points, [10](#)
remove_points_polygon, [11](#)
response_curve_bart, [11](#)
run_glossa, [12](#)

variable_importance, [13](#)