Package: glossa (via r-universe)

October 29, 2026

Type Package

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

Version 1.2.5

Description A user-friendly 'shiny' application for Bayesian machine learning analysis of marine species distributions. GLOSSA (Global Ocean Species Spatio-temporal Analysis) uses Bayesian Additive Regression Trees (BART; Chipman, George, and McCulloch (2010) <doi: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. Optionally, pseudo-absences can be generated within the environmental space using the external package 'flexsdm' (not on CRAN), which can be downloaded from <https://github.com/sjevelazco/flexsdm>; this functionality is used conditionally when available and all core features work without it.

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 (>= 4.1.0), shiny

Imports automap, blockCV, dbarts, dplyr, DT, GeoThinneR, ggplot2, htmltools, leaflet, markdown, pROC, sf, shinyWidgets, sparkline, svglite, terra, tidyterra, waiter, zip

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

Config/testthat/edition 3

Encoding UTF-8

Language en-US

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Jorge Mestre-Tomás [aut, cre] (ORCID: <https://orcid.org/0000-0002-8983-3417>), Alba Fuster-Alonso [aut] (ORCID: <https://orcid.org/0000-0002-7283-291X>)

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

Repository https://cran.r-universe.dev

Date/Publication 2026-10-29 21:15:08 UTC

RemoteUrl https://github.com/cran/glossa

RemoteRef HEAD

RemoteSha 8a60b3249cee03b860c02ab50962f75679f15290

Contents

buffer_polygon . . . 3
clean_coordinates . . . 3
contBoyce . . . 4
create_coords_layer . . . 6
cross_validate_model . . . 7
evaluation_metrics . . . 7
extract_noNA_cov_values . . . 8
fit_bart_model . . . 9
generate_cv_folds . . . 9
generate_pa_buffer_out . . . 10
generate_pa_env_space_flexsdm . . . 11
generate_pa_random . . . 12
generate_pa_target_group . . . 13
generate_pseudo_absences . . . 14
glossa_analysis . . . 15
invert_polygon . . . 17
layer_mask . . . 18
pa_optimal_cutoff . . . 18
plot_cv_folds_points . . . 19
predict_bart . . . 19
remove_duplicate_points . . . 20
remove_points_polygon . . . 20
response_curve_bart . . . 21
run_glossa . . . 22
variable_importance . . . 23

Index **24**

`buffer_polygon` *Enlarge/Buffer a Polygon*

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` *Clean Coordinates of Presence/Absence Data*

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,
  thinning_method = NULL,
  thinning_value = 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).
thinning_method	Character; spatial thinning method to apply to occurrence data. Options are ‘c("None", "Distance", "Grid", "Precision")‘. See ‘GeoThinneR‘ package for details.
thinning_value	Numeric; value used for thinning depending on the selected method: distance in meters (‘Distance‘), grid resolution in degrees (‘Grid‘), or decimal precision (‘Precision‘).
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.

---

contBoyce *Continuous Boyce Index (CBI) with weighting*

Description

This function is a copy from the ‘contBoyce()‘ function from the ‘enmSdm‘ R package. This function calculates the continuous Boyce index (CBI), a measure of model accuracy for presence-only test data. This version uses multiple, overlapping windows, in contrast to `link{contBoyce2x}`, which covers each point by at most two windows.

Usage

```
contBoyce(
  pres,
  contrast,
  presWeight = rep(1, length(pres)),
  contrastWeight = rep(1, length(contrast)),
  numBins = 101,
  binWidth = 0.1,
  autoWindow = TRUE,
  method = "spearman",
  dropZeros = TRUE,
  na.rm = FALSE,
  ...
)
```

Arguments

`pres`	Numeric vector. Predicted values at presence sites.
`contrast`	Numeric vector. Predicted values at background sites.
`presWeight`	Numeric vector same length as `pres`. Relative weights of presence sites. The default is to assign each presence a weight of 1.
`contrastWeight`	Numeric vector same length as `contrast`. Relative weights of background sites. The default is to assign each presence a weight of 1.
`numBins`	Positive integer. Number of (overlapping) bins into which to divide predictions.
`binWidth`	Positive numeric value < 1. Size of a bin. Each bin will be `binWidth * (max - min)`. If `autoWindow` is `FALSE` (the default) then `min` is 0 and `max` is 1. If `autoWindow` is `TRUE` then `min` and `max` are the maximum and minimum value of all predictions in the background and presence sets (i.e., not necessarily 0 and 1).
`autoWindow`	Logical. If `FALSE` calculate bin boundaries starting at 0 and ending at 1 + epsilon (where epsilon is a very small number to assure inclusion of cases that equal 1 exactly). If `TRUE` (default) then calculate bin boundaries starting at minimum predicted value and ending at maximum predicted value.
`method`	Character. Type of correlation to calculate. The default is `'spearman'`, the Spearman rank correlation coefficient used by Boyce et al. (2002) and Hirzel et al. (2006), which is the "traditional" CBI. In contrast, `'pearson'` or `'kendall'` can be used instead. See `cor` for more details.
`dropZeros`	Logical. If `TRUE` then drop all bins in which the frequency of presences is 0.
`na.rm`	Logical. If `TRUE` then remove any presences and associated weights and background predictions and associated weights with `NA`s.
`...`	Other arguments (not used).

Details

CBI is the Spearman rank correlation coefficient between the proportion of sites in each prediction class and the expected proportion of predictions in each prediction class based on the proportion of

the landscape that is in that class. The index ranges from -1 to 1. Values >0 indicate the model's output is positively correlated with the true probability of presence. Values <0 indicate it is negatively correlated with the true probability of presence.

Value

Numeric value.

Note

This function is directly copied from the 'enmSdm' package.

References

Boyce, M.S., Vernier, P.R., Nielsen, S.E., and Schmiegelow, F.K.A. 2002. Evaluating resource selection functions. *Ecological Modeling* 157:281-300. doi:10.1016/S03043800(02)002004

Hirzel, A.H., Le Lay, G., Helfer, V., Randon, C., and Guisan, A. 2006. Evaluating the ability of habitat suitability models to predict species presences. *Ecological Modeling* 199:142-152. doi:10.1016/j.ecolmodel.2006.05.017

---

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.

cross_validate_model *Cross-validation for BART model*

Description

This function performs 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

```
cross_validate_model(data, folds, predictor_cols = NULL, seed = NULL)
```

Arguments

data	Data frame with a column (named 'pa') indicating presence (1) or absence (0) and columns for the predictor variables.
folds	A vector of fold assignments (same length as 'data').
predictor_cols	Optional; a character vector of column names to be used as predictors. If NULL, all columns except 'pa' will be used.
seed	Optional; random seed.

Value

A list with:

metrics 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.

predictions Data frame with observed, predicted, probability, and fold assignment per test instance.

evaluation_metrics *Evaluation metrics for model predictions*

Description

Computes a set of performance metrics (e.g., AUC, TSS, CBI) based on observed and predicted values.

Usage

```
evaluation_metrics(df, na.rm = TRUE, method = "spearman")
```

Arguments

df	A data.frame with columns: ‘observed‘ (0/1), ‘predicted‘ (0/1), ‘probability‘ (numeric).
na.rm	Logical. Whether to remove rows with NA values.
method	Correlation method for CBI ("spearman", "pearson", or "kendall").

Value

A named list or data.frame with evaluation metrics.

---

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 *Fit a BART Model Using Environmental Covariate Layers*

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.
...	Additional arguments passed to ‘dbarts::bart()‘.

Value

A BART model object.

generate_cv_folds *Generate cross-validation folds*

Description

Creates cross-validation fold assignments for presence-absence or presence-only data, supporting three types of strategies: k-fold, spatial blocks (through blockCV R package), and temporal blocks.

Usage

```
generate_cv_folds(
  data,
  method = "k-fold",
  block_method = "predictors_autocorrelation",
  block_size = NULL,
  k = 10,
  predictor_raster = NULL,
  model_residuals = NULL,
  coords = c("decimalLongitude", "decimalLatitude")
)
```

Arguments

`data`	A ‘data.frame‘ with at least presence-absence data (‘pa‘), coordinates, and optionally a ‘timestamp‘.
`method`	The cross-validation strategy. One of: ‘"k-fold"‘, ‘"spatial_blocks"‘, ‘"temporal_blocks"‘.
`block_method`	For spatial blocks, how to determine block size. One of: ‘"residuals_autocorrelation"‘, ‘"predictors_autocorrelation"‘, ‘"manual"‘.
`block_size`	Numeric. Manual block size in meters (used if ‘block_method = "manual"‘).
`k`	Integer. Number of folds to generate.
`predictor_raster`	A ‘terra::SpatRaster‘ used for estimating spatial autocorrelation (only needed if ‘block_method = "predictors_autocorrelation"‘).
`model_residuals`	A ‘data.frame‘ with residuals and coordinates (only needed if ‘block_method = "residuals_autocorrelation"‘).
`coords`	A character vector of length 2 indicating the longitude and latitude column names.

Value

A list with the following elements:

folds A vector of fold assignments (one per row in ‘data‘).

method The CV method used.

block_method The spatial block size method (if applicable).

block_size The estimated or manual block size (in meters), if spatial blocks were used.

`generate_pa_buffer_out` *Generate Pseudo-Absences Using Buffer-Out Strategy*

Description

This function generates pseudo-absences outside a buffer around presence points but within the convex hull of those points. This prevents spatial overlap while preserving geographic realism.

Usage

```
generate_pa_buffer_out(
  presences,
  raster_stack,
  predictor_variables,
  coords = c("decimalLongitude", "decimalLatitude"),
  pa_buffer_distance = 0.5,
```

```
  ratio = 1,
  attempts = 100,
  seed = NULL
)
```

Arguments

presences	Data frame containing presence points.
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")'.
pa_buffer_distance	Numeric; buffer radius in degrees around each presence. Default is 0.5.
ratio	Ratio of pseudo-absences to presences (default 1 = balanced).
attempts	Integer specifying the number of attempts to generate exact pseudo-absences. Defaults to 100.
seed	Optional seed for reproducibility.

Value

A data frame of pseudo-absences with coordinates, timestamp, 'pa = 0', and covariate values.

---

generate_pa_env_space_flexsdm

Generate Environmental-space Pseudo-Absences via flexsdm (per temporal stratum)

Description

Uses flexsdm::sample_pseudoabs(method = c("env_const", env = predictors) within each timestamp stratum so that pseudo-absences match the temporal distribution of presences. Extracts covariate values for the sampled coordinates and returns a data.frame with the same predictor columns.

Usage

```
generate_pa_env_space_flexsdm(
  presences,
  raster_stack,
  predictor_variables,
  coords = c("decimalLongitude", "decimalLatitude"),
  ratio = 1,
  attempts = 100,
  seed = NULL
)
```

Arguments

presences	Data frame containing presence points.
raster_stack	'SpatRaster' object containing covariate data. Uses continuous predictor only and samples per timestamp.
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")'.
ratio	Ratio of pseudo-absences to presences (default 1 = balanced).
attempts	Integer specifying the number of attempts to generate exact pseudo-absences. Defaults to 100.
seed	Optional seed for reproducibility.

Value

A data frame of pseudo-absences with coordinates, timestamp, 'pa = 0', and covariate values.

---

generate_pa_random *Generate Random Pseudo-Absences*

Description

This function generates pseudo-absence points randomly across the study area (random background), optionally applying spatial thinning to match presence filtering strategy.

Usage

```
generate_pa_random(
  presences,
  study_area,
  raster_stack,
  predictor_variables,
  coords = c("decimalLongitude", "decimalLatitude"),
  ratio = 1,
  attempts = 100,
  seed = NULL
)
```

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")‘.
ratio	Ratio of pseudo-absences to presences (default 1 = balanced).
attempts	Integer specifying the number of attempts to generate exact pseudo-absences. Defaults to 100.
seed	Optional random seed.

Value

Data frame containing pseudo-absence points with coordinates, timestamp, pa = 0, and covariates.

`generate_pa_target_group` *Generate Pseudo-Absences Using Target-Group Background*

Description

Generate Pseudo-Absences Using Target-Group Background

Usage

```
generate_pa_target_group(
  presences,
  target_group_points,
  study_area,
  raster_stack,
  predictor_variables,
  coords = c("decimalLongitude", "decimalLatitude"),
  ratio = 1,
  attempts = 100,
  seed = NULL
)
```

Arguments

presences	Data frame containing presence points.
target_group_points	Data frame of all sampling locations (target group).
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")‘.
ratio	Ratio of pseudo-absences to presences (default 1 = balanced).

attempts	Integer specifying the number of attempts to generate exact pseudo-absences. Defaults to 100.
seed	Optional random seed.

Value

Data frame containing pseudo-absence points with coordinates, timestamp, pa = 0, and covariates.

`generate_pseudo_absences` *Generate Pseudo-Absence Points Using Different Methods Based on Presence Points, Covariates, and Study Area Polygon*

Description

Wrapper function for pseudo-absence generation methods, such as background random points, target-group, and using buffer area.

Usage

```
generate_pseudo_absences(
  method = c("random", "target_group", "buffer_out", "env_space_flexsdm"),
  presences,
  raster_stack,
  predictor_variables,
  study_area = NULL,
  target_group_points = NULL,
  coords = c("decimalLongitude", "decimalLatitude"),
  pa_buffer_distance = 0.5,
  ratio = 1,
  attempts = 100,
  seed = NULL
)
```

Arguments

method	Character; one of "random", "target_group", "buffer_out", or "env_space_flexsdm".
presences	Data frame of presence points with coordinates and timestamp.
raster_stack	SpatRaster of covariates.
predictor_variables	Character vector of selected predictors.
study_area	Optional sf polygon (used for clipping).
target_group_points	Optional data frame of sampling points (for target-group).
coords	Vector of coordinate column names.

pa_buffer_distance
: Numeric; buffer radius in degrees around each presence. Default is 0.5.

ratio
: Ratio of pseudo-absences to presences.

attempts
: Max attempts to fulfill sample size.

seed
: Optional seed for reproducibility.

Value

A data frame of pseudo-absence points (pa = 0) with covariates.

---

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,
  thinning_method = NULL,
  thinning_value = NULL,
  scale_layers = FALSE,
  buffer = NULL,
  native_range = NULL,
  suitable_habitat = NULL,
  other_analysis = NULL,
  model_args = list(),
  cv_methods = NULL,
  cv_folds = 5,
  cv_block_source = "residuals_autocorrelation",
  cv_block_size = NULL,
  pseudoabsence_method = "random",
  pa_ratio = 1,
  target_group_points = NULL,
  pa_buffer_distance = NULL,
  seed = NA,
  waiter = NULL
)
```

Arguments

pa_data	A list of data frames containing presence-absence data including ‘decimalLongitude‘, ‘decimalLatitude‘, ‘timestamp‘, and ‘pa‘ columns.
fit_layers	A ZIP file with the raster files containing model fitting environmental layers formatted as explained in the website documentation.
proj_files	A list of ZIP file paths containing environmental layers for projection scenarios.
study_area_poly	A spatial polygon defining the study area.
predictor_variables	A list of the predictor variables to be used in the analysis for each occurrence dataset.
thinning_method	A character specifying the spatial thinning method to apply to occurrence data. Options are ‘c("none", "distance", "grid", "precision")‘. See ‘GeoThinneR‘ package for details.
thinning_value	A numeric value used for thinning depending on the selected method: distance in meters (‘distance‘), grid resolution in degrees (‘grid‘), or decimal precision (‘precision‘).
scale_layers	Logical; if ‘TRUE‘, covariate layers will be standardize (z-score) based on fit layers.
buffer	Buffer value or distance in decimal degrees (arc_degrees) for buffering the study area polygon.
native_range	A vector of scenarios ‘c(’fit_layers’, ’projections’)‘ where native range modeling should be performed.
suitable_habitat	A vector of scenarios ‘c(’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’‘).
model_args	A named list of additional arguments passed to the modeling function (e.g., ‘dbarts::bart‘). This allows users to fine-tune model parameters such as ‘ntree‘ or ‘k‘. These are passed internally via ‘...‘ and must match the arguments of the selected model function.
cv_methods	A vector of the cross-validation strategies to perform. One or multiple of ‘"k-fold"‘, ‘"spatial_blocks"‘, ‘"temporal_blocks"‘.
cv_folds	Integer indicating the number of folds to generate.
cv_block_source	For spatial blocks, how to determine block size. One of: ‘"residuals_autocorrelation"‘, ‘"predictors_autocorrelation"‘, ‘"manual"‘.
cv_block_size	Numeric block size in meters (used if ‘cv_block_source = "manual"‘).
pseudoabsence_method	Method for generating pseudo-absences. One of "random", "target_group", "buffer_out", or "env_space_flexsdm".

pa_ratio	Ratio of pseudo-absences to presences (pseudo-absence:presences).
target_group_points	Optional data frame for sampling points for target-group method.
pa_buffer_distance	Numeric buffer radius in degrees around each presence. Default is NULL.
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 *Invert a Polygon*

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 *Apply Polygon Mask to Raster Layers*

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 *Optimal Cutoff for Presence-Absence Prediction*

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.

plot_cv_folds_points *Plot cross-validation fold assignments*

Description

Plot cross-validation fold assignments

Usage

```
plot_cv_folds_points(data, polygon = NULL)
```

Arguments

data	Dataframe with columns: 'decimalLongitude', 'decimalLatitude', 'pa' and 'fold'.
polygon	An sf object representing the inverted study area.

Value

A ggplot object showing point color-coded by cv fold and shaped by presence/absence.

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 *Run GLOSSA Shiny App*

Description

This function launches the GLOSSA Shiny web application.

Usage

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

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.
clear_global_env	Logical. If TRUE, clears the global environment after the app exits.

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.

Note

Use ‘clear_global_env = TRUE‘ cautiously, as it removes all objects from your R environment after the app exits.

Examples

```
if(interactive()) {
run_glossa()
run_glossa(clear_global_env = TRUE)  # clears all global objects
}
```

`variable_importance` *Variable Importance in BART Model*

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, y, x, cutoff = 0, n_repeats = 10, seed = NULL)
```

Arguments

`bart_model`	A BART model object.
`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.
`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, 3

clean_coordinates, 3
contBoyce, 4
cor, *5*
create_coords_layer, 6
cross_validate_model, 7

evaluation_metrics, 7
extract_noNA_cov_values, 8

fit_bart_model, 9

generate_cv_folds, 9
generate_pa_buffer_out, 10
generate_pa_env_space_flexsdm, 11
generate_pa_random, 12
generate_pa_target_group, 13
generate_pseudo_absences, 14
glossa_analysis, 15

invert_polygon, 17

layer_mask, 18

pa_optimal_cutoff, 18
plot_cv_folds_points, 19
predict_bart, 19

remove_duplicate_points, 20
remove_points_polygon, 20
response_curve_bart, 21
run_glossa, 22

variable_importance, 23