Package 'envi'

August 2, 2021

Type Package

Title Environmental Interpolation using Spatial Kernel Density Estimation

Version 0.1.10

Date 2021-08-02

Maintainer Ian D. Buller <ian.buller@alumni.emory.edu>

Description Estimates an ecological niche using occurrence data, covariates, and kernel density-based estimation methods. For a single species with presence and absence data, the 'envi' package uses the spatial relative risk function that is estimated using the 'sparr' package. Details about the 'sparr' package methods can be found in the tutorial: Davies et al. (2018) <doi:10.1002/sim.7577>. Details about kernel density estimation can be found in J. F. Bithell (1990) <doi:10.1002/sim.4780090616>. More information about relative risk functions using kernel density estimation can be found in J. F. Bithell (1991) <doi:10.1002/sim.4780101112>.

License Apache License (>= 2.0)

Encoding UTF-8

RoxygenNote 7.1.1

Depends R (>= 3.5.0), spatstat.geom, spatstat.core, spatstat.linnet, spatstat (>= 2.0-0)

Imports concaveman, cvAUC, doFuture, doRNG, fields, foreach, future, graphics, grDevices, iterators, maptools, pls, raster, rgeos, ROCR, sp, sparr, stats

Suggests R.rsp, spatstat.data, spelling, testthat

VignetteBuilder R.rsp

Language en-US

URL https://github.com/Waller-SUSAN/envi

BugReports https://github.com/Waller-SUSAN/envi/issues

NeedsCompilation no

Author Ian D. Buller [aut, cre, cph] (<https://orcid.org/0000-0001-9477-8582>), Lance A. Waller [ctb, ths] (<https://orcid.org/0000-0001-5002-8886>), Emory University [cph]

Repository CRAN

Date/Publication 2021-08-02 20:10:08 UTC

R topics documented:

`envi-package` *The envi Package: Environmental Interpolation using Spatial Kernel Density Estimation*

Description

Estimates an ecological niche model using occurrence data, covariates, and kernel density-based estimation methods.

Details

For a single species with presence and absence data, the 'envi' package uses the spatial relative risk function that is estimated using the 'sparr' package. Details about the 'sparr' package methods can be found in the tutorial: Davies et al. (2018) doi: 10.1002/sim.7577. Details about kernel density estimation can be found in J. F. Bithell (1990) doi: 10.1002/sim.4780090616. More information about relative risk functions using kernel density estimation (KDE) can be found in J. F. Bithell (1991) doi: 10.1002/sim.4780101112.

This package provides a function to estimate the ecological niche for a single species with presence and absence data. The 'envi' package also provides some sensitivity and visualization tools for the estimated ecological niche, its predicted spatial distribution, and prediction diagnostics.

Key content of the 'envi' package include:

Ecological Niche Model

`lrren` Estimates the ecological niche for a single species with presence/absence data, two covariates, and the spatial relative risk function. Provide functionality to predict the spatial distribution of the estimated ecological niche in geographic space and prepare internal k-fold cross-validation data.

Sensitivity Analysis

perlrren Iteratively estimates the ecological niche for a single species with spatially perturbed ("jittered") presence/absence data, two covariates, and the spatial relative risk function. Various radii for the spatial perturbation can be specified.

Data Visualization

plot_obs Visualizes the lrren output, specifically the estimated ecological niche in a space with dimensions as the two specified covariates in the model.

plot_predict Visualizes the lrren output, specifically the predicted spatial distribution of the ecological niche.

plot_cv Visualizes the lrren output, specifically two prediction diagnostics (area under the receiver operating characteristic curve and precision-recall curve).

plot_perturb Visualizes the perlrren output, specifically four summary statistics of the iterations including mean log relative risk, standard deviation of the log relative risk, mean p-value, and proportion of iterations the p-value was significant based on an alpha-level threshold. Also has functionality to predict the spatial distribution of the summary statistics.

Dependencies

The 'envi' package relies heavily upon sparr, spatstat.geom, and raster. For a single species (presence/absence data), the spatial relative risk function uses the risk function. Cross-validation is can be performed in parallel using the future, doFuture, doRNG, and foreach packages. Spatial perturbation is performed using the rjitter function. Basic visualizations rely on the plot.ppp and image.plot functions.

Author(s)

Ian D. Buller
Environmental Health Sciences, Emory University, Atlanta, Georgia, USA.

Maintainer: I.D.B. <ian.buller@alumni.emory.edu>

`lrren` *Ecological niche model using a log relative risk surface*

Description

Estimate the ecological niche of a single species with presence/absence data and two covariates. Predict the ecological niche in geographic space.

Usage

```
lrren(
  obs_locs,
  predict = FALSE,
  predict_locs = NULL,
  conserve = TRUE,
```

```
  alpha = 0.05,
  p_correct = "none",
  cv = FALSE,
  kfold = 10,
  balance = FALSE,
  parallel = FALSE,
  n_core = 2,
  poly_buffer = NULL,
  obs_window = NULL,
  verbose = FALSE,
  ...
)
```

Arguments

obs_locs	Input data frame of presence and absence observations with six (6) features (columns): 1) ID, 2) longitude, 3) latitude, 4) presence/absence binary variable, 5) covariate 1 as x-coordinate, 6) covariate 2 as y-coordinate.
predict	Logical. If TRUE, will predict the ecological niche in geographic space. If FALSE (the default), will not predict.
predict_locs	Input data frame of prediction locations with 4 features (columns): 1) longitude, 2) latitude, 3) covariate 1 as x-coordinate, 4) covariate 2 as y-coordinate. The covariates must be the same as those included in obs_locs.
conserve	Logical. If TRUE (the default), the ecological niche will be estimated within a concave hull around the locations in obs_locs. If FALSE, the ecological niche will be estimated within a concave hull around the locations in predict_locs.
alpha	Numeric. The two-tailed alpha level for significance threshold (default is 0.05).
p_correct	Optional. Character string specifying whether to apply a correction for multiple comparisons including a False Discovery Rate p_correct = "FDR", a Sidak correction p_correct = "uncorrelated Sidak", and a Bonferroni correction p_correct = "uncorrelated Bonferroni". If p_correct = "none" (the default), then no correction is applied.
cv	Logical. If TRUE, will calculate prediction diagnostics using internal k-fold cross-validation. If FALSE (the default), will not.
kfold	Integer. Specify the number of folds using in the internal cross-validation. Default is 10.
balance	Logical. If TRUE, the prevalence within each k-fold will be 0.50 by undersampling absence locations (assumes absence data are more frequent). If FALSE (the default), the prevalence within each k-fold will match the prevalence in obs_locs.
parallel	Logical. If TRUE, will execute the function in parallel. If FALSE (the default), will not execute the function in parallel.
n_core	Optional. Integer specifying the number of CPU cores on current host to use for parallelization (the default is 2 cores).

poly_buffer	Optional. Specify a custom distance (in same units as covariates) to add to window within which the ecological niche is estimated. The default is 1/100th of the smallest range among the two covariates.
obs_window	Optional. Specify a custom window of class 'owin' within which to estimate the ecological niche. The default computes a concave hull around the data specified in conserve.
verbose	Logical. If TRUE (the default), will print function progress during execution. If FALSE, will not print.
...	Arguments passed to risk to select bandwidth, edge correction, and resolution.

Details

This function estimates the ecological niche of a single species (presence/absence data), or the presence of one species relative to another, using two covariates, will predict the ecological niche into a geographic area, and prepare k-fold cross-validation data sets for prediction diagnostics.

The function uses the risk function to estimate the spatial relative risk function and forces risk(tolerate == TRUE) in order to calculate asymptotic p-values. The estimated ecological niche can be visualized using the plot_obs function.

If predict = TRUE, this function will predict ecological niche at every location specified with predict_locs with best performance if predict_locs are gridded locations in the same study area as the observations in obs_locs - a version of environmental interpolation. The predicted spatial distribution of the estimated ecological niche can be visualized using the plot_predict function.

If cv = TRUE, this function will prepare k-fold cross-validation data sets for prediction diagnostics. The sample size of each fold depends on the number of folds set with kfold. If balance = TRUE, the sample size of each fold will be frequency of presence locations divided by number of folds times two. If balance = FALSE, the sample size of each fold will be frequency of all observed locations divided by number of folds. The cross-validation can be performed in parallel if parallel = TRUE using the future, doFuture, doRNG, and foreach packages. Two diagnostics (area under the receiver operating characteristic curve and precision-recall curve) can be visualized using the plot_cv function.

The obs_window argument may be useful to specify a 'known' window for the ecological niche (e.g., a convex hull around observed locations).

This function has functionality for a correction for multiple testing. If p_correct = "FDR", calculates a False Discovery Rate by Benjamini and Hochberg. If p_correct = "Sidak", calculates a Sidak correction. If p_correct = "Bonferroni", calculates a Bonferroni correction. If p_correct = "none" (the default), then the function does not account for multiple testing and uses the uncorrected alpha level. See the internal pval_correct function documentation for more details.

Value

An object of class 'list'. This is a named list with the following components:

out An object of class 'list' for the estimated ecological niche.

dat An object of class 'data.frame', returns obs_locs that are used in the accompanying plotting functions.

`p_critical` A numeric value for the critical p-value used for significance tests.

The returned `out` is a named list with the following components:

`obs` An object of class 'rrs' for the spatial relative risk.

`presence` An object of class 'ppp' for the presence locations.

`absence` An object of class 'ppp' for the absence locations.

`outer_poly` An object of class 'matrix' for the coordinates of the concave hull around the observation locations.

`inner_poly` An object of class 'matrix' for the coordinates of the concave hull around the observation locations. Same as `outer_poly`.

If `predict = TRUE`, the returned `out` has additional components:

`outer_poly` An object of class 'matrix' for the coordinates of the concave hull around the prediction locations.

`prediction` An object of class 'matrix' for the coordinates of the concave hull around the prediction locations.

If `cv = TRUE`, the returned object of class 'list' has an additional named list `cv` with the following components:

`cv_predictions_rr` A list of length `kfold` with values of the log relative risk surface at each point randomly selected in a cross-validation fold.

`cv_labels` A list of length `kfold` with a binary value of presence (1) or absence (0) for each point randomly selected in a cross-validation fold.

Examples

```
if (interactive()) {
  set.seed(1234) # for reproducibility

# Using the 'bei' and 'bei.extra' data within {spatstat.data}

# Covariate data (centered and scaled)
  elev <- spatstat.data::bei.extra[[1]]
  grad <- spatstat.data::bei.extra[[2]]
  elev$v <- scale(elev)
  grad$v <- scale(grad)
  elev_raster <- raster::raster(elev)
  grad_raster <- raster::raster(grad)

# Presence data
  presence <- spatstat.data::bei
  spatstat.geom::marks(presence) <- data.frame("presence" = rep(1, presence$n),
                                               "lon" = presence$x,
                                               "lat" = presence$y)
  spatstat.geom::marks(presence)$elev <- elev[presence]
  spatstat.geom::marks(presence)$grad <- grad[presence]
```

```
# (Pseudo-)Absence data
  absence <- spatstat.core::rpoispp(0.008, win = elev)
  spatstat.geom::marks(absence) <- data.frame("presence" = rep(0, absence$n),
                                              "lon" = absence$x,
                                              "lat" = absence$y)
  spatstat.geom::marks(absence)$elev <- elev[absence]
  spatstat.geom::marks(absence)$grad <- grad[absence]

# Combine into readable format
  obs_locs <- spatstat.geom::superimpose(presence, absence, check = FALSE)
  obs_locs <- spatstat.geom::marks(obs_locs)
  obs_locs$id <- seq(1, nrow(obs_locs), 1)
  obs_locs <- obs_locs[ , c(6, 2, 3, 1, 4, 5)]

# Prediction Data
  predict_locs <- data.frame(raster::rasterToPoints(elev_raster))
  predict_locs$layer2 <- raster::extract(grad_raster, predict_locs[, 1:2])

# Run lrren
  test_lrren <- lrren(obs_locs = obs_locs,
                      predict_locs = predict_locs,
                      predict = TRUE,
                      cv = TRUE)
}
```

---

perlrren *Spatially perturb an ecological niche model that uses a log relative risk surface*

Description

Estimates the ecological niche of a single species with presence/absence data and two covariates, iteratively, by randomly perturbing ('jittering') the coordinates of observations.

Usage

```
perlrren(
  obs_ppp,
  covariates,
  predict = TRUE,
  predict_locs = NULL,
  radii = NULL,
  n_sim = 2,
  alpha = 0.05,
  p_correct = "none",
  parallel = FALSE,
  n_core = 2,
  verbose = FALSE,
```

```
  ...
)
```

Arguments

obs_ppp	Input object of class 'ppp' a marked point pattern of presence and absence observations with 5 (five) features (columns): 1) ID, 2) longitude, 3) latitude, 4) presence/absence binary variable, 5) ordinal ID for spatial perturbation.
covariates	Input object of class 'imlist' of 2 (two) covariates within the same spatial window and in the same coordinate reference system as obs_ppp.
predict	Logical. If TRUE (the default), will predict the ecological niche in geographic space. If FALSE, will not predict.
predict_locs	Input data frame of prediction locations with 4 features (columns): 1) longitude, 2) latitude, 3) covariate 1 as x-coordinate, 4) covariate 2 as y-coordinate. If unspecified (the default), automatically computed from an 'im' object within covariates.
radii	Vector of length equal to the number of levels of ordinal ID in obs_ppp. Specifies the radii of the spatial perturbation at each level in units equivalent to the coordinate reference system of obs_ppp.
n_sim	Integer, specifying the number of simulation iterations to perform.
alpha	Numeric. The two-tailed alpha level for significance threshold (default is 0.05).
p_correct	Optional. Character string specifying whether to apply a correction for multiple comparisons including a False Discovery Rate p_correct = "FDR", a Sidak correction p_correct = "uncorrelated Sidak", and a Bonferroni correction p_correct = "uncorrelated Bonferroni". If p_correct = "none" (the default), then no correction is applied.
parallel	Logical. If TRUE, will execute the function in parallel. If FALSE (the default), will not execute the function in parallel.
n_core	Optional. Integer specifying the number of CPU cores on current host to use for parallelization (the default is 2 cores).
verbose	Logical. If TRUE (the default), will print function progress during execution. If FALSE, will not print.
...	Arguments passed to lrren.

Details

This function performs a sensitivity analysis of an ecological niche model of a single species (presence/absence data), or the presence of one species relative to another, that uses two covariates. The observation locations (presence and absence data) are randomly spatially perturbed (i.e., "jittered") uniformly within a circular disc of a specified radius centered at their recorded location using the rjitter function. This method simulates the spatial uncertainty of observations, how that may affect the covariate values at each observation (i.e., misclassification error), and the estimated ecological niche based on the two specified covariates. Observations can be grouped into categories of uncertainty of class 'factor' and can vary by degrees of uncertainty specified using the radii argument.

The function iteratively estimates the ecological niche using the lrren function and computes four summary statistics at every grid cell (i.e., knot) of the estimated surface: 1) mean of the log relative risk, 2) standard deviation of the log relative risk, 3) mean of the asymptotically normal p-value, and 4) proportion of iterations were statistically significant based on a two-tailed alpha-level threshold (argument alpha). The process can be performed in parallel if parallel = TRUE using the future, doFuture, doRNG, and foreach packages. The computed surfaces can be visualized using the plot_perturb function. If predict = TRUE, this function will predict the four summary statistics at every location specified with predict_locs and can also be visualized using the plot_perturb function.

For more information about the spatial perturbation, please refer to the rjitter function documentation.

The function has functionality for a correction for multiple testing. If p_correct = "FDR", calculates a False Discovery Rate by Benjamini and Hochberg. If p_correct = "Sidak", calculates a Sidak correction. If p_correct = "Bonferroni", calculates a Bonferroni correction. If p_correct = "none" (the default), then the function does not account for multiple testing and uses the uncorrected alpha level. See the internal pval_correct function documentation for more details.

Value

An object of class "list". This is a named list with the following components:

sim An object of class 'list' for the summary statistics of the iterative ecological niche.

predict An object of class 'ppp' a marked point pattern with summary statistics for the iterative ecological niche in geographic space.

The returned sim is a named list with the following components:

lrr_mean An object of class 'im' for the mean log relative risk surface.

lrr_sd An object of class 'im' for the standard deviation of log relative risk surface.

pval_mean An object of class 'im' for the mean p-value surface.

pval_prop An object of class 'im' for the proportion of iterations were statistically significant surface.

alpha_median A numeric value of the median critical p-value across all iterations.

If predict = FALSE, the returned predict is empty. If predict = TRUE, the returned predict is an object of class 'ppp' a marked point pattern with the following features:

x Values for x-coordinate in geographic space (e.g., longitude).

y Values for y-coordinate in geographic space (e.g., latitude).

v Values for x-coordinate in covariate space.

z Values for x-coordinate in covariate space.

lrr_mean Values for the mean log relative risk surface.

lrr_sd Values for the standard deviation of log relative risk surface.

pval_mean Values for the mean p-value surface.

pval_prop Values for the proportion of iterations were statistically significant surface.

Examples

```
if (interactive()) {
  set.seed(1234) # for reproducibility

# Using the 'bei' and 'bei.extra' data within {spatstat.data}

# Covariate data (centered and scaled)
  ims <- spatstat.data::bei.extra
  ims[[1]]$v <- scale(ims[[1]]$v)
  ims[[2]]$v <- scale(ims[[2]]$v)

# Presence data
  presence <- spatstat.data::bei
  spatstat.geom::marks(presence) <- data.frame("presence" = rep(1, presence$n),
                                               "lon" = presence$x,
                                               "lat" = presence$y)

# (Pseudo-)Absence data
  absence <- spatstat.core::rpoispp(0.008, win = ims[[1]])
  spatstat.geom::marks(absence) <- data.frame("presence" = rep(0, absence$n),
                                               "lon" = absence$x,
                                               "lat" = absence$y)
# Combine into readable format
  obs_locs <- spatstat.geom::superimpose(presence, absence, check = FALSE)
  spatstat.geom::marks(obs_locs)$id <- seq(1, obs_locs$n, 1)
  spatstat.geom::marks(obs_locs) <- spatstat.geom::marks(obs_locs)[ , c(4, 2, 3, 1)]

# Specify categories for varying degrees of spatial uncertainty
## Creates three groups
  spatstat.geom::marks(obs_locs)$levels <- as.factor(stats::rpois(obs_locs$n,
                                                                  lambda = 0.05))

# Run perlrren
  test_perlrren <- perlrren(obs_ppp = obs_locs,
                            covariates = ims,
                            radii = c(10, 100, 500),
                            n_sim = 10)
}
```

`plot_cv` *Visualizations for the prediction diagnostics of an estimated ecological niche*

Description

Create multiple plots of output from the `lrren` function, specifically for the internal k-fold cross-validation diagnostics.

Usage

```
plot_cv(input, alpha = 0.05)
```

Arguments

input	An object of class 'list' from the lrren function.
alpha	Numeric. The two-tailed alpha level for significance threshold (default is 0.05).

Value

This function produces two plots: 1) area under the receiver operating characteristic curve, and 2) precision-recall curve. Each plot shows predictions for the log relative risk surface. The red-colored lines are the average curves.

Examples

```
if (interactive()) {
  set.seed(1234) # for reproducibility

# Using the 'bei' and 'bei.extra' data within {spatstat.data}

# Covariate data (centered and scaled)
  elev <- spatstat.data::bei.extra[[1]]
  grad <- spatstat.data::bei.extra[[2]]
  elev$v <- scale(elev)
  grad$v <- scale(grad)
  elev_raster <- raster::raster(elev)
  grad_raster <- raster::raster(grad)

# Presence data
  presence <- spatstat.data::bei
  spatstat.geom::marks(presence) <- data.frame("presence" = rep(1, presence$n),
                                               "lon" = presence$x,
                                               "lat" = presence$y)
  spatstat.geom::marks(presence)$elev <- elev[presence]
  spatstat.geom::marks(presence)$grad <- grad[presence]

# (Pseudo-)Absence data
  absence <- spatstat.core::rpoispp(0.008, win = elev)
  spatstat.geom::marks(absence) <- data.frame("presence" = rep(0, absence$n),
                                              "lon" = absence$x,
                                              "lat" = absence$y)
  spatstat.geom::marks(absence)$elev <- elev[absence]
  spatstat.geom::marks(absence)$grad <- grad[absence]

# Combine into readable format
  obs_locs <- spatstat.geom::superimpose(presence, absence, check = FALSE)
  obs_locs <- spatstat.geom::marks(obs_locs)
  obs_locs$id <- seq(1, nrow(obs_locs), 1)
  obs_locs <- obs_locs[ , c(6, 2, 3, 1, 4, 5)]
```

```
# Prediction Data
  predict_locs <- data.frame(raster::rasterToPoints(elev_raster))
  predict_locs$layer2 <- raster::extract(grad_raster, predict_locs[, 1:2])

# Run lrren
  test_lrren <- lrren(obs_locs = obs_locs,
                      predict_locs = predict_locs,
                      predict = TRUE,
                      cv = TRUE)

# Run plot_cv
  plot_cv(input = test_lrren)
}
```

---

plot_obs *Visualizations for an estimated ecological niche in covariate space*

Description

Create multiple plots of output from the lrren function, specifically for the observation data and estimated ecological niche.

Usage

```
plot_obs(
  input,
  plot_cols = c("#8B3A3A", "#CCCCCC", "#0000CD"),
  alpha = input$p_critical,
  lower_lrr = NULL,
  upper_lrr = NULL,
  digits = 1,
  ...
)
```

Arguments

input	An object of class 'list' from the lrren function.
plot_cols	Character string of length three (3) specifying the colors for plotting: 1) presence, 2) neither, and 3) absence. The default colors in hex are c("#8B3A3A","#CCCCCC","#0000CD") or c("indianred4","grey80","blue3").
alpha	Optional, numeric. The two-tailed alpha level for significance threshold (default is the p_critical value imported from input).
lower_lrr	Optional, numeric. Lower cut-off value for the log relative risk value in the color key (typically a negative value). The default is no limit and the color key will include the minimum value of the log relative risk surface.

upper_lrr	Optional, numeric. Upper cut-off value for the log relative risk value in the color key (typically a positive value). The default is no limit and the color key will include the maximum value of the log relative risk surface.
digits	Optional, integer. The number of significant digits for the color key labels using the round function (default is 1).
...	Arguments passed to plot.ppp and image.plot for additional graphical features.

Value

This function produces three plots in a two-dimensional space where the axes are the two specified covariates: 1) observation locations by group, 2) log relative risk surface, and 3) significant p-value surface.

Examples

```
if (interactive()) {
  set.seed(1234) # for reproducibility

# Using the 'bei' and 'bei.extra' data within {spatstat.data}

# Covariate data (centered and scaled)
  elev <- spatstat.data::bei.extra[[1]]
  grad <- spatstat.data::bei.extra[[2]]
  elev$v <- scale(elev)
  grad$v <- scale(grad)
  elev_raster <- raster::raster(elev)
  grad_raster <- raster::raster(grad)

# Presence data
  presence <- spatstat.data::bei
  spatstat.geom::marks(presence) <- data.frame("presence" = rep(1, presence$n),
                                               "lon" = presence$x,
                                               "lat" = presence$y)
  spatstat.geom::marks(presence)$elev <- elev[presence]
  spatstat.geom::marks(presence)$grad <- grad[presence]

# (Pseudo-)Absence data
  absence <- spatstat.core::rpoispp(0.008, win = elev)
  spatstat.geom::marks(absence) <- data.frame("presence" = rep(0, absence$n),
                                              "lon" = absence$x,
                                              "lat" = absence$y)
  spatstat.geom::marks(absence)$elev <- elev[absence]
  spatstat.geom::marks(absence)$grad <- grad[absence]

# Combine into readable format
  obs_locs <- spatstat.geom::superimpose(presence, absence, check = FALSE)
  obs_locs <- spatstat.geom::marks(obs_locs)
  obs_locs$id <- seq(1, nrow(obs_locs), 1)
  obs_locs <- obs_locs[ , c(6, 2, 3, 1, 4, 5)]
```

```
# Prediction Data
  predict_locs <- data.frame(raster::rasterToPoints(elev_raster))
  predict_locs$layer2 <- raster::extract(grad_raster, predict_locs[, 1:2])

# Run lrren
  test_lrren <- lrren(obs_locs = obs_locs,
                      predict_locs = predict_locs,
                      predict = TRUE,
                      cv = TRUE)

# Run plot_obs
  plot_obs(input = test_lrren)
}
```

`plot_perturb` *Visualizations for a simulated ecological niche after iteratively perturbing the observation coordinates*

Description

Create multiple plots of output from the `perlrren` function, specifically for the four summary statistics in covariate space and geographic space.

Usage

```
plot_perturb(
  input,
  predict = TRUE,
  mean_cols = c("#8B3A3A", "#CCCCCC", "#0000CD"),
  var_cols = c("#E5E5E5", "#1A1A1A"),
  cov_labs = c("V1", "V2"),
  cref0 = "EPSG:4326",
  cref1 = NULL,
  lower_lrr = NULL,
  upper_lrr = NULL,
  upper_sd = NULL,
  digits = 1,
  ...
)
```

Arguments

`input`	An object of class 'list' from the `perlrren` function.
`predict`	Logical. If TRUE (the default), will visualize the four summary statistics in geographic space. If FALSE, will not.

mean_cols	Character string of length three (3) specifying the colors for plots with a divergent color palette: 1) presence, 2) neither, and 3) absence. The default colors in hex are `c("#8B3A3A","#CCCCCC","#0000CD")` or `c("indianred4","grey80","blue3")`.
var_cols	Character string of length two (2) specifying the colors for plots with a sequential color palette from low to high values. The default colors in hex are `c("#E5E5E5","#1A1A1A")` or `c("grey90","grey10")`.
cov_labs	Character string of length two (2) specifying the x- and y-axis labels in plots of the ecological niche in covariate space. The default values are generic `c("V1","V2")`.
cref0	Character. The Coordinate Reference System (CRS) for the x- and y-coordinates in geographic space. The default is WGS84 `"EPSG:4326"`.
cref1	Optional, character. The Coordinate Reference System (CRS) to spatially project the x- and y-coordinates in geographic space.
lower_lrr	Optional, numeric. Lower cut-off value for the log relative risk value in the color key (typically a negative value). The default is no limit and the color key will include the minimum value of the log relative risk surface.
upper_lrr	Optional, numeric. Upper cut-off value for the log relative risk value in the color key (typically a positive value). The default is no limit and the color key will include the maximum value of the log relative risk surface.
upper_sd	Optional, numeric. Upper cut-off value for the standard deviation of log relative risk value in the color key. The default is no limit and the color key will include the maximum value of the standard deviation surface.
digits	Optional, integer. The number of significant digits for the color key labels using the `round` function (default is 1).
...	Arguments passed to `image.plot` for additional graphical features.

Value

This function produces four plots in a two-dimensional space where the axes are the two specified covariates: 1) mean of the log relative risk, 2) standard deviation of the log relative risk, 3) mean of the asymptotically normal p-value, and 4) proportion of iterations were statistically significant based on a two-tailed alpha-level threshold. If `predict = TRUE`, this function produces an additional four plots of the summary statistics above in a two-dimensional geographic space where the axes are longitude and latitude.

Examples

```
if (interactive()) {
  set.seed(1234) # for reproducibility

# Using the 'bei' and 'bei.extra' data within {spatstat.data}

# Covariate data (centered and scaled)
  ims <- spatstat.data::bei.extra
  ims[[1]]$v <- scale(ims[[1]]$v)
  ims[[2]]$v <- scale(ims[[2]]$v)

# Presence data
```

```
  presence <- spatstat.data::bei
  spatstat.geom::marks(presence) <- data.frame("presence" = rep(1, presence$n),
                                               "lon" = presence$x,
                                               "lat" = presence$y)

# (Pseudo-)Absence data
  absence <- spatstat.core::rpoispp(0.008, win = ims[[1]])
  spatstat.geom::marks(absence) <- data.frame("presence" = rep(0, absence$n),
                                               "lon" = absence$x,
                                               "lat" = absence$y)
# Combine into readable format
  obs_locs <- spatstat.geom::superimpose(presence, absence, check = FALSE)
  spatstat.geom::marks(obs_locs)$id <- seq(1, obs_locs$n, 1)
  spatstat.geom::marks(obs_locs) <- spatstat.geom::marks(obs_locs)[ , c(4, 2, 3, 1)]

# Specify categories for varying degrees of spatial uncertainty
## Creates three groups
  spatstat.geom::marks(obs_locs)$levels <- as.factor(stats::rpois(obs_locs$n,
                                                                  lambda = 0.05))

# Run perlrren
  test_perlrren <- perlrren(obs_ppp = obs_locs,
                            covariates = ims,
                            radii = c(10, 100, 500),
                            n_sim = 10)

# Run plot_perturb
  plot_perturb(input = test_perlrren)
}
```

---

plot_predict *Visualizations for a predicted ecological niche in geographic space*

Description

Create multiple plots of output from the lrren function, specifically for the predicted values of the ecological niche at geographic coordinates.

Usage

```
plot_predict(
  input,
  plot_cols = c("#8B3A3A", "#CCCCCC", "#0000CD", "#FFFF00"),
  alpha = input$p_critical,
  cref0 = "EPSG:4326",
  cref1 = NULL,
  lower_lrr = NULL,
  upper_lrr = NULL,
```

```
  digits = 1,
  ...
)
```

Arguments

input	An object of class 'list' from the lrren function.
plot_cols	Character string of length four (4) specifying the colors for plotting: 1) presence, 2) neither, 3) absence, and 4) NA values. The default colors in hex are c("#8B3A3A","#CCCCCC","#0000CD" "#FFFF00") or c("indianred4","grey80","blue3","yellow"
alpha	Optional, numeric. The two-tailed alpha level for significance threshold (default is the p_critical value imported from input).
cref0	Character. The Coordinate Reference System (CRS) for the x- and y-coordinates in geographic space. The default is WGS84 "EPSG:4326".
cref1	Optional, character. The Coordinate Reference System (CRS) to spatially project the x- and y-coordinates in geographic space.
lower_lrr	Optional, numeric. Lower cut-off value for the log relative risk value in the color key (typically a negative value). The default is no limit and the color key will include the minimum value of the log relative risk surface.
upper_lrr	Optional, numeric. Upper cut-off value for the log relative risk value in the color key (typically a positive value). The default is no limit and the color key will include the maximum value of the log relative risk surface.
digits	Optional, integer. The number of significant digits for the color key labels using the round function (default is 1).
...	Arguments passed to image.plot for additional graphical features.

Value

This function produces two plots in a two-dimensional space where the axes are geographic coordinates (e.g., longitude and latitude): 1) predicted log relative risk, and 2) significant p-values.

Examples

```
if (interactive()) {
  set.seed(1234) # for reproducibility

# Using the 'bei' and 'bei.extra' data within {spatstat.data}

# Covariate data (centered and scaled)
  elev <- spatstat.data::bei.extra[[1]]
  grad <- spatstat.data::bei.extra[[2]]
  elev$v <- scale(elev)
  grad$v <- scale(grad)
  elev_raster <- raster::raster(elev)
  grad_raster <- raster::raster(grad)

# Presence data
  presence <- spatstat.data::bei
```

```
  spatstat.geom::marks(presence) <- data.frame("presence" = rep(1, presence$n),
                                               "lon" = presence$x,
                                               "lat" = presence$y)
  spatstat.geom::marks(presence)$elev <- elev[presence]
  spatstat.geom::marks(presence)$grad <- grad[presence]

# (Pseudo-)Absence data
  absence <- spatstat.core::rpoispp(0.008, win = elev)
  spatstat.geom::marks(absence) <- data.frame("presence" = rep(0, absence$n),
                                              "lon" = absence$x,
                                              "lat" = absence$y)
  spatstat.geom::marks(absence)$elev <- elev[absence]
  spatstat.geom::marks(absence)$grad <- grad[absence]

# Combine into readable format
  obs_locs <- spatstat.geom::superimpose(presence, absence, check = FALSE)
  obs_locs <- spatstat.geom::marks(obs_locs)
  obs_locs$id <- seq(1, nrow(obs_locs), 1)
  obs_locs <- obs_locs[ , c(6, 2, 3, 1, 4, 5)]

# Prediction Data
  predict_locs <- data.frame(raster::rasterToPoints(elev_raster))
  predict_locs$layer2 <- raster::extract(grad_raster, predict_locs[, 1:2])

# Run lrren
  test_lrren <- lrren(obs_locs = obs_locs,
                      predict_locs = predict_locs,
                      predict = TRUE,
                      cv = TRUE)

# Run plot_predict
  plot_predict(input = test_lrren, cref0 = "EPSG:5472")
}
```

Index