Package 'tenm'

July 23, 2024

Version 0.5.1

Title Temporal Ecological Niche Models

Description Implements methods and functions to calibrate time-specific niche models (multi-temporal calibration), letting users execute a strict calibration and selection process of niche models based on ellipsoids, as well as functions to project the potential distribution in the present and in global change scenarios.The 'tenm' package has functions to recover information that may be lost or overlooked while applying a data curation protocol. This curation involves preserving occurrences that may appear spatially redundant (occurring in the same pixel) but originate from different time periods. A novel aspect of this package is that it might reconstruct the fundamental niche more accurately than mono-calibrated approaches. The theoretical background of the package can be found in Peterson et al. (2011)<doi:10.5860/CHOICE.49-6266>.

License GPL-3

URL https://luismurao.github.io/tenm/

BugReports https://github.com/luismurao/tenm/issues

Imports MASS, terra (> 1.7.5), sf, purrr, dplyr, stringr, rgl (> 1.2), future, tidyr, furrr, lubridate, methods

RoxygenNote 7.3.1

Encoding UTF-8

Depends R (>= 4.1)

LazyData true

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Luis Osorio-Olvera [aut, cre] (<https://orcid.org/0000-0003-0701-5398>), Miguel Hernández [aut] (<https://orcid.org/0000-0002-6086-3460>), Rusby G. Contreras-Díaz [aut] (<https://orcid.org/0000-0002-0569-8984>), Xavier Chiappa-Carrara [aut] (<https://orcid.org/0000-0002-1708-2095>),

Fernanda Rosales-Ramos [aut] (<https://orcid.org/0009-0004-7805-4735>),
Mariana Munguía-Carrara [aut] (<https://orcid.org/0000-0003-3514-3397>),
Oliver López-Corona [aut] (<https://orcid.org/0000-0002-2926-7791>),
Townsend Peterson [ctb] (<https://orcid.org/0000-0003-0243-2379>),
Jorge Soberón [ctb] (<https://orcid.org/0000-0003-2160-4148>)

Maintainer Luis Osorio-Olvera <luismurao@gmail.com>

Repository CRAN

Date/Publication 2024-07-23 00:30:01 UTC

Contents

abronia 2
bg_by_date 3
cells2samp 4
clean_dup 6
clean_dup_by_date 8
colors 9
correlation_finder 10
cov_center 11
ellipsoid_omr 12
ellipsoid_projection 14
ellipsoid_selection 16
ex_by_date 18
inEllipsoid 19
metaras 21
plot_ellipsoid 21
predict,sp.temporal.selection-method 23
pROC 26
sp.temporal.bg-class 27
sp.temporal.env-class 28
sp.temporal.modeling-class 28
sp.temporal.selection-class 29
sp_temporal_data 29
tdf2swd 31
tenm_selection 32

Index **35**

`abronia` *Occurrence records of Abronia graminea*

Description

A dataset containing occurrence records for Abronia graminea. The data was downloaded from GBIF (GBIF, 2022).

Usage

```
abronia
```

Format

A data frame with 106 rows and 5 variables:

species Scientific name of the species

decimalLongitude Longitude

decimalLatitude Latitude

year Observation year

gbif_doi DOI id for citing the dataset ...

Source

GBIF.org (22 February 2022) GBIF Occurrence Download doi:10.15468/dl.teyjm9

---

bg_by_date *Function to obtain environmental background organized by date*

Description

Function to retrieve background data from occurrence records. The background data is organized as a function of the dated environmental data.

Usage

```
bg_by_date(
  this_species,
  buffer_ngbs = NULL,
  buffer_distance = 1000,
  n_bg = 50000,
  process_ngbs_by = 100
)
```

Arguments

this_species	An object of class sp.temporal.env representing species occurrence data organized by date. See ex_by_date.
buffer_ngbs	Number of pixel neighbors used to build the buffer around each occurrence point.
buffer_distance	Distance (in the same units as raster layers) used to create a buffer around occurrence points to sample background data.
n_bg	Number of background points to sample.
process_ngbs_by	Numeric parameter to improve memory management. It process neighbor cells by a quantity specified by the user.

Details

This function retrieves background data around species occurrence points, sampled based on the dated environmental data provided in `this_species`. Background points are sampled within a buffer around each occurrence point. The function returns an object of class sp.temporal.bg, which contains background data organized by date. This object is the input of the function `tenm_selection`.

Value

An object of class sp.temporal.bg containing background data organized by date. The object is a list with the following components:

- "bg_df": A data.frame with columns for longitude, latitude, year, layer_date, layer_path, cell_ids_year, and environmental information.
- Other metadata relevant to background sampling.

Examples

```
library(tenm)
data("abronia")
tempora_layers_dir <- system.file("extdata/bio",package = "tenm")
abt <- tenm::sp_temporal_data(occs = abronia,
                              longitude = "decimalLongitude",
                              latitude = "decimalLatitude",
                              sp_date_var = "year",
                              occ_date_format="y",
                              layers_date_format= "y",
                              layers_by_date_dir = tempora_layers_dir,
                              layers_ext="*.tif$")
abtc <- tenm::clean_dup_by_date(abt,threshold = 10/60)
#This code is for running in parallel
future::plan("multisession",workers=2)
abex <- tenm::ex_by_date(this_species = abtc,train_prop=0.7)
abbg <- tenm::bg_by_date(this_species = abex,
                         buffer_ngbs=10,n_bg=50000)
future::plan("sequential")
```

---

cells2samp *Helper function to randomly select cell IDs for generating environmental background data.*

Description

This function returns pixel IDs to be sampled for generating environmental background data around species occurrence points.

Usage

```
cells2samp(
  data,
  longitude,
  latitude,
  cell_ids = NULL,
  buffer_ngbs = 2,
  raster_mask,
  process_ngbs_by = 10,
  n_bg = 50000,
  progress = TRUE
)
```

Arguments

data	A data.frame containing longitude and latitude data of occurrence points.
longitude	A character vector specifying the column name of longitude in 'data'.
latitude	A character vector specifying the column name of latitude in 'data'.
cell_ids	A numeric vector indicating the IDs of cells that serve as geographic centers for buffers. Default is NULL.
buffer_ngbs	Number of neighboring pixels around occurrence points used to build the buffer for sampling.
raster_mask	An object of class SpatRaster used to obtain pixel IDs.
process_ngbs_by	Numeric parameter to improve memory management. It process neighbor cells by a quantity specified by the user.
n_bg	Number of background pixels to sample.
progress	Logical. If TRUE, show computation progress.

Value

A numeric vector of cell IDs to be sampled for environmental background data.

Examples

```
# cells to sample
data(abronia)
temporal_layer <- system.file("extdata/bio/2016/bio_01.tif",package = "tenm")
raster_mask <- terra::rast(temporal_layer)
set.seed(123)
samp_01 <- tenm::cells2samp(data = abronia,
                            longitude = "decimalLongitude",
                            latitude = "decimalLatitude",
                            cell_ids = NULL,
                            buffer_ngbs = 4,
                            raster_mask = raster_mask,
                            process_ngbs_by = 10,
```

```
                            n_bg = 50000,
                            progress =TRUE)
# Generete a sample using pixel IDs
samp_02 <- tenm::cells2samp(data = abronia,
                            longitude = NULL,
                            latitude = NULL,
                            cell_ids = c(256,290,326),
                            buffer_ngbs = 4,
                            raster_mask = raster_mask,
                            process_ngbs_by = 10,
                            n_bg = 50000,
                            progress =TRUE)
```

---

clean_dup *Function to thin longitude and latitude data*

Description

Cleans up duplicated or redundant occurrence records that present overlapping longitude and latitude coordinates. Thinning can be performed using either a geographical distance threshold or a pixel neighborhood approach.

Usage

```
clean_dup(
  data,
  longitude,
  latitude,
  threshold = 0,
  by_mask = FALSE,
  raster_mask = NULL,
  n_ngbs = 0
)
```

Arguments

data	A data.frame with longitude and latitude of occurrence records.
longitude	A character vector indicating the column name of the "longitude" variable.
latitude	A character vector indicating the column name of the "latitude" variable.
threshold	A numeric value representing the distance threshold between coordinates to be considered duplicates. Units depend on whether by_mask is T or F. If T, the user needs to specify the number of pixels that define the neighborhood of duplicates (see n_ngbs parameter).
by_mask	Logical. If T, the thinning process will use a raster layer as a mask for defining distance in pixel units.

raster_mask — An object of class SpatRaster that serves as a reference to thin the occurrence data. Required if by_mask is T.

n_ngbs — Number of pixels used to define the neighborhood matrix that helps determine which occurrences are duplicates:

- 0 removes occurrences within the same pixel, keeping one.
- 1 considers duplicates all occurrences within a distance of one pixel.
- n considers duplicates all occurrences within a distance of n pixels.

Details

This function cleans up duplicated occurrences based on the specified distance threshold. If by_mask is T, the distance is interpreted as pixel distance using the provided raster_mask; otherwise, it is interpreted as geographic distance.

Value

Returns a data.frame with cleaned occurrence records, excluding duplicates based on the specified criteria.

Examples

```
data(abronia)
tempora_layers_dir <- system.file("extdata/bio",package = "tenm")
tenm_mask <- terra::rast(file.path(tempora_layers_dir,"1939/bio_01.tif"))
# Clean duplicates without raster mask (just by distance threshold)
# First check the number of occurrence records
print(nrow(abronia))
# Clean duplicated records using a distance of ~ 18 km (0.1666667 grades)
ab_1 <- tenm::clean_dup(data =abronia,
                        longitude = "decimalLongitude",
                        latitude = "decimalLatitude",
                        threshold = terra::res(tenm_mask),
                        by_mask = FALSE,
                        raster_mask = NULL)
# Check number of records
print(nrow(ab_1))
# Clean duplicates using a raster mask
ab_2 <- tenm::clean_dup(data =abronia,
                        longitude = "decimalLongitude",
                        latitude = "decimalLatitude",
                        threshold = terra::res(tenm_mask)[1],
                        by_mask = TRUE,
                        raster_mask = tenm_mask,
                        n_ngbs = 1)
# Check number of records
print(nrow(ab_2))
```

clean_dup_by_date *Function to thin occurrence data Cleans up duplicated longitude and latitude data by year using a specified distance threshold. The distance can be specified as a geographic distance or, if a raster_mask is provided, as a pixel distance.*

Description

Function to thin occurrence data Cleans up duplicated longitude and latitude data by year using a specified distance threshold. The distance can be specified as a geographic distance or, if a raster_mask is provided, as a pixel distance.

Usage

```
clean_dup_by_date(
  this_species,
  threshold,
  by_mask = FALSE,
  raster_mask = NULL,
  n_ngbs = 0
)
```

Arguments

this_species	An object of class sp.temporal.modeling representing species occurrence data organized by date. See sp_temporal_data.
threshold	A numeric value representing the distance threshold between coordinates to be considered duplicates. Units depend on whether by_mask is TRUE or FALSE. If TRUE, the user needs to specify the number of pixels that define the neighborhood of duplicates (see n_ngbs parameter).
by_mask	Logical. If TRUE, the thinning process will use a raster layer as a mask for defining distance in pixel units.
raster_mask	An object of class SpatRaster that serves as a reference to thin the occurrence data. Required if by_mask is TRUE.
n_ngbs	Number of pixels used to define the neighborhood matrix that helps determine which occurrences are duplicates: • 0 removes occurrences within the same pixel, keeping one. • 1 considers duplicates all occurrences within a distance of one pixel. • n considers duplicates all occurrences within a distance of n pixels.

Details

This function is based on clean_dup. It cleans up duplicated occurrences based on the specified threshold. If by_mask is TRUE, the distance is interpreted as pixel distance using the provided raster_mask; otherwise, it is interpreted as geographic distance.

Value

An object of class sp.temporal.modeling containing a temporal data.frame with cleaned occurrence data, including columns for longitude, latitude, date variable, layers_dates, and layers_path.

Examples

```
library(tenm)
data("abronia")
tempora_layers_dir <- system.file("extdata/bio",package = "tenm")
tenm_mask <- terra::rast(file.path(tempora_layers_dir,"1939/bio_01.tif"))
# Clean duplicates without raster mask (just by distance threshold)
abt <- tenm::sp_temporal_data(occs = abronia,
                              longitude = "decimalLongitude",
                              latitude = "decimalLatitude",
                              sp_date_var = "year",
                              occ_date_format="y",
                              layers_date_format= "y",
                              layers_by_date_dir = tempora_layers_dir,
                              layers_ext="*.tif$")
abtc1 <- tenm::clean_dup_by_date(abt,threshold = terra::res(tenm_mask)[1])
# Check number of records
print(nrow(abtc1$temporal_df))
# Clean duplicates using a raster mask
abtc2 <- tenm::clean_dup_by_date(this_species = abt,
                                 by_mask = TRUE,
                                 threshold = terra::res(tenm_mask)[1],
                                 raster_mask = tenm_mask,
                                 n_ngbs = 0)
# Check number of records
print(nrow(abtc2$temporal_df))

abtc3 <- tenm::clean_dup_by_date(this_species = abt,
                                 by_mask = TRUE,
                                 threshold = terra::res(tenm_mask)[1],
                                 raster_mask = tenm_mask,
                                 n_ngbs = 2)
# Check number of records
print(nrow(abtc3$temporal_df))
```

---

colors *Colors for plotting*

Description

A string vector of colors for plotting the vignette example.

Usage

```
colors
```

Format

An object of class `character` of length 40.

`correlation_finder` *Function to find strong correlations within environmental predictors*

Description

This function identifies variables with strong correlations based on a specified threshold.

Usage

```
correlation_finder(
  environmental_data,
  method = "spearman",
  threshold,
  verbose = TRUE
)
```

Arguments

`environmental_data`	A matrix or data.frame containing environmental data.
`method`	Method used to estimate the correlation matrix. Possible options include "spearman" (Spearman's rank correlation), "pearson" (Pearson's correlation), or "kendall" (Kendall's tau correlation).
`threshold`	Correlation threshold value. Variables with absolute correlation values greater than or equal to this threshold are considered strongly correlated.
`verbose`	Logical. If `TRUE`, prints verbose output detailing correlations.

Value

A list with two elements:

- `not_correlated_vars`: A vector containing names of variables that are not strongly correlated.
- `correlation_values`: A list with correlation values for all pairs of variables.

Examples

```
library(tenm)
data("abronia")
tempora_layers_dir <- system.file("extdata/bio",package = "tenm")
abt <- tenm::sp_temporal_data(occs = abronia,
                              longitude = "decimalLongitude",
                              latitude = "decimalLatitude",
```

```
                                sp_date_var = "year",
                                occ_date_format="y",
                                layers_date_format= "y",
                                layers_by_date_dir = tempora_layers_dir,
                                layers_ext="*.tif$")
abtc <- tenm::clean_dup_by_date(abt,threshold = 10/60)
future::plan("multisession",workers=2)
abex <- tenm::ex_by_date(abtc,train_prop=0.7)
future::plan("sequential")
envdata <- abex$env_data[,-ncol(abex$env_data)]
ecors <- tenm::correlation_finder(environmental_data =envdata,
                                  method="spearman",
                                  threshold = 0.7 )
```

`cov_center` *Function to compute the covariance matrix of an ellipsoid niche model.*

Description

Computes the covariance matrix, niche centroid, volume, and other ellipsoid parameter based on the values of niche variables from occurrence points.

Usage

```
cov_center(data, mve = TRUE, level, vars = NULL)
```

Arguments

`data`	A data.frame or matrix containing numeric values of variables used to model the niche.
`mve`	Logical. If `TRUE`, computes a minimum volume ellipsoid using the `cov.mve` function from the MASS package. If `FALSE`, uses the covariance matrix of the input data.
`level`	Proportion of data to be used for computing the ellipsoid, applicable when mve is `TRUE`.
`vars`	Vector specifying column indexes or names of variables in the input data used to fit the ellipsoid model.

Value

A list containing the following components:

- `centroid`: Centroid (mean vector) of the ellipsoid.
- `covariance_matrix`: Covariance matrix based on the input data.

- `volume`: Volume of the ellipsoid.
- `semi_axes_lengths`: Lengths of semi-axes of the ellipsoid.
- `axis_coordinates`: Coordinates of ellipsoid axes.

Examples

```
library(tenm)
data("abronia")
tempora_layers_dir <- system.file("extdata/bio",package = "tenm")
abt <- tenm::sp_temporal_data(occs = abronia,
                              longitude = "decimalLongitude",
                              latitude = "decimalLatitude",
                              sp_date_var = "year",
                              occ_date_format="y",
                              layers_date_format= "y",
                              layers_by_date_dir = tempora_layers_dir,
                              layers_ext="*.tif$")
abtc <- tenm::clean_dup_by_date(abt,threshold = 10/60)
future::plan("multisession",workers=2)
abex <- tenm::ex_by_date(abtc,train_prop=0.7)
future::plan("sequential")
mod <- tenm::cov_center(data = abex$env_data,
                        mve = TRUE,
                        level = 0.975,
                        vars = c("bio_05","bio_06","bio_12"))
# Print model parameters
print(mod)
```

---

ellipsoid_omr *Compute omission rate and statistical metrics for ellipsoid models.*

Description

Computes omission rate and statistical metrics for ellipsoid models using environmental data.

Usage

```
ellipsoid_omr(
  env_data,
  env_test = NULL,
  env_bg,
  cf_level,
  mve = TRUE,
  proc = FALSE,
  proc_iter = 100,
  rseed = TRUE
)
```

Arguments

env_data	A data frame containing the environmental data used for modeling.
env_test	A data frame with environmental testing data. Default is NULL. If provided, the selection process includes p-values from a binomial test.
env_bg	Environmental data sampled from the calibration area to compute the approximated prevalence of the model.
cf_level	Proportion of points to be included in the ellipsoids. Equivalent to the error (E) proposed by Peterson et al. (2008). doi:10.1016/j.ecolmodel.2007.11.008.
mve	Logical. If TRUE, computes a minimum volume ellipsoid using cov.rob from the MASS package. If FALSE, uses the covariance matrix of the input data.
proc	Logical. If TRUE, performs a partial ROC test.
proc_iter	Numeric. Total number of iterations for the partial ROC bootstrap.
rseed	Logical. If TRUE, sets a random seed for the partial ROC bootstrap. Default is TRUE.

Value

A data.frame with the following columns:

- "fitted_vars": Names of variables that were fitted.
- "nvars": Number of fitted variables
- "om_rate_train": Omission rate of the training data.
- "non_pred_train_ids": Row IDs of non-predicted training data.
- "om_rate_test"': Omission rate of the testing data.
- "non_pred_test_ids": Row IDs of non-predicted testing data.
- "bg_prevalence": Approximated prevalence of the model (see details).
- "pval_bin": p-value of the binomial test.
- "pval_proc": p-value of the partial ROC test.
- "env_bg_paucratio": Environmental partial AUC ratio value.
- "env_bg_auc": Environmental AUC value.

Examples

```
library(tenm)
data("abronia")
tempora_layers_dir <- system.file("extdata/bio",package = "tenm")
abt <- tenm::sp_temporal_data(occs = abronia,
                              longitude = "decimalLongitude",
                              latitude = "decimalLatitude",
                              sp_date_var = "year",
                              occ_date_format="y",
                              layers_date_format= "y",
                              layers_by_date_dir = tempora_layers_dir,
                              layers_ext="*.tif$")
```

```
abtc <- tenm::clean_dup_by_date(abt,threshold = 10/60)
#This code is for running in parallel
future::plan("multisession",workers=2)
abex <- tenm::ex_by_date(this_species = abtc,train_prop=0.7)
abbg <- tenm::bg_by_date(this_species = abex,
                         buffer_ngbs=10,n_bg=50000)
future::plan("sequential")
edata <- abex$env_data
etrain <- edata[edata$trian_test=="Train",c("bio_05","bio_06","bio_12")]
etest <- edata[edata$trian_test=="Test",c("bio_05","bio_06","bio_12")]
bg <- abbg$env_bg[,c("bio_05","bio_06","bio_12")]
eor <- ellipsoid_omr(env_data=etrain,env_test=etest,env_bg=bg,
                     cf_level=0.975,proc=TRUE)
eor
```

`ellipsoid_projection` *ellipsoid_projection: function to project an ellipsoid model*

Description

Function to project an ellipsoid model using the shape matrix (covariance matrix) of the niche variables.

Usage

```
ellipsoid_projection(
  envlayers,
  centroid,
  covar,
  level = 0.95,
  output = "suitability",
  plot = TRUE,
  size,
  xlab1 = "niche var 1",
  ylab1 = "niche var 2",
  zlab1 = "S",
  alpha = 0.1,
  ...
)
```

Arguments

`envlayers`	A SpatRaster object of the niche variables.
`centroid`	A vector with the values of the centers of the ellipsoid (see `cov_center`).
`covar`	The shape matrix (covariance) of the ellipsoid (see `cov_center`).
`level`	The proportion of points to be included inside the ellipsoid

output	The output distance: two possible values "suitability" or "mahalanobis". By default the function uses "suitability".
plot	Logical If TRUE a plot of the niche will be shown.
size	The size of the points of the niche plot.
xlab1	For x label for 2-dimensional histogram
ylab1	For y label for 2-dimensional histogram
zlab1	For z label for 2-dimensional histogram
alpha	Control the transparency of the 3-dimensional ellipsoid
...	Arguments passed to plot3d function from rgl

Value

Returns a SpatRaster of suitability values.

Examples

```
library(tenm)
data("abronia")
tempora_layers_dir <- system.file("extdata/bio",package = "tenm")
abt <- tenm::sp_temporal_data(occs = abronia,
                              longitude = "decimalLongitude",
                              latitude = "decimalLatitude",
                              sp_date_var = "year",
                              occ_date_format="y",
                              layers_date_format= "y",
                              layers_by_date_dir = tempora_layers_dir,
                              layers_ext="*.tif$")
abtc <- tenm::clean_dup_by_date(abt,threshold = 10/60)
future::plan("multisession",workers=2)
abex <- tenm::ex_by_date(this_species = abtc,train_prop=0.7)
abbg <- tenm::bg_by_date(this_species = abex,
                         buffer_ngbs=10,n_bg=50000)
future::plan("sequential")
mod <- tenm::cov_center(data = abex$env_data,
                        mve = TRUE,
                        level = 0.975,
                        vars = c("bio_05","bio_06","bio_12"))
layers_path <-   list.files(file.path(tempora_layers_dir,
                                      "2016"),
                            pattern = ".tif$",full.names = TRUE)
elayers <- terra::rast(layers_path)
nmod <- ellipsoid_projection(envlayers = elayers[[names(mod$centroid)]],
                             centroid = mod$centroid,
                             covar = mod$covariance,
                             level = 0.99999,
                             output = "suitability",
                             size = 3,
                             plot = TRUE)
```

`ellipsoid_selection` *ellipsoid_selection: Performs models selection for ellipsoid models*

Description

The function performs model selection for ellipsoid models using three criteria: a) the omission rate, b) the significance of partial ROC and binomial tests and c) the AUC value.

Usage

```
ellipsoid_selection(
  env_train,
  env_test = NULL,
  env_vars,
  nvarstest,
  level = 0.95,
  mve = TRUE,
  env_bg = NULL,
  omr_criteria,
  parallel = FALSE,
  ncores = NULL,
  comp_each = 100,
  proc = FALSE,
  proc_iter = 100,
  rseed = TRUE
)
```

Arguments

`env_train`	A data frame with the environmental training data.
`env_test`	A data frame with the environmental testing data. Default is NULL.
`env_vars`	A vector with the names of environmental variables used in the selection process. To help choosing which variables to use see `correlation_finder`.
`nvarstest`	A vector indicating the number of variables to fit the ellipsoids during model selection.
`level`	Proportion of points to be included in the ellipsoids, equivalent to the error (E) proposed by Peterson et al. (2008).
`mve`	Logical. If `TRUE`, a minimum volume ellipsoid will be computed. using `cov.rob` from **MASS**. If `FALSE`, the covariance matrix of the input data will be used.
`env_bg`	Environmental data to compute the approximated prevalence of the model, should be a sample of the environmental layers of the calibration area.
`omr_criteria`	Omission rate criteria: the allowable omission rate for the selection process. Default is NULL (see details).
`parallel`	Logical. If `TRUE`, computations will run in parallel. Default is `F`.

ncores	Number of cores to use for parallel processing. Default uses all available cores minus one.
comp_each	Number of models to run in each job in parallel computation. Default is 100.
proc	Logical. If TRUE, a partial ROC test will be run.
proc_iter	Numeric. Total iterations for the partial ROC bootstrap.
rseed	Logical. If TRUE, set a random seed for partial ROC bootstrap. Default is TRUE.

Details

Model selection occurs in environmental space (E-space). For each variable combination specified in nvarstest, the omission rate (omr) in E-space is computed using `inEllipsoid` function. Results are ordered by omr of the testing data. If env_bg is provided, an estimated prevalence is computed and results are additionally ordered by partial AUC. Model selection can be run in parallel. For more details and examples go to `ellipsoid_omr` help.

Value

A data.frame with the following columns:

- "fitted_vars": Names of variables that were fitted.
- "nvars": Number of fitted variables
- "om_rate_train": Omission rate of the training data.
- "non_pred_train_ids": Row IDs of non-predicted training data.
- "om_rate_test"': Omission rate of the testing data.
- "non_pred_test_ids": Row IDs of non-predicted testing data.
- "bg_prevalence": Approximated prevalence of the model (see details).
- "pval_bin": p-value of the binomial test.
- "pval_proc": p-value of the partial ROC test.
- "env_bg_paucratio": Environmental partial AUC ratio value.
- "env_bg_auc": Environmental AUC value.
- "mean_omr_train_test": Mean value of omission rates (train and test).
- "rank_by_omr_train_test": Rank value of importance in model selection by omission rate.
- "rank_omr_aucratio": Rank value by AUC ratio.

Author(s)

Luis Osorio-Olvera luismurao@gmail.com

References

Peterson, A.T. et al. (2008) Rethinking receiver operating characteristic analysis applications in ecological niche modeling. Ecol. Modell. 213, 63–72. doi:10.1016/j.ecolmodel.2007.11.008

Examples

```
library(tenm)
data("abronia")
tempora_layers_dir <- system.file("extdata/bio",package = "tenm")
abt <- tenm::sp_temporal_data(occs = abronia,
                              longitude = "decimalLongitude",
                              latitude = "decimalLatitude",
                              sp_date_var = "year",
                              occ_date_format="y",
                              layers_date_format= "y",
                              layers_by_date_dir = tempora_layers_dir,
                              layers_ext="*.tif$")
abtc <- tenm::clean_dup_by_date(abt,threshold = 10/60)
future::plan("multisession",workers=2)
abex <- tenm::ex_by_date(this_species = abtc,train_prop=0.7)
abbg <- tenm::bg_by_date(this_species = abex,
                         buffer_ngbs=10,n_bg=50000)
future::plan("sequential")
varcorrs <- tenm::correlation_finder(environmental_data =
                                     abex$env_data[,-ncol(abex$env_data)],
                                     method = "spearman",
                                     threshold = 0.8,
                                     verbose = FALSE)
edata <- abex$env_data
etrain <- edata[edata$trian_test=="Train",] |> data.frame()
etest <- edata[edata$trian_test=="Test",] |> data.frame()
bg <- abbg$env_bg
res1 <- tenm::ellipsoid_selection(env_train = etrain,
                                  env_test = etest,
                                  env_vars = varcorrs$descriptors,
                                  nvarstest = 3,
                                  level = 0.975,
                                  mve = TRUE,
                                  env_bg = bg,
                                  omr_criteria = 0.1,
                                  parallel = FALSE,proc = TRUE)
head(res1)
```

---

ex_by_date *Extract environmental data by date*

Description

Function to extract environmental data by date. It generates training and testing datasets using a random partition with a specified proportion.

Usage

```
ex_by_date(this_species, train_prop = 0.7)
```

Arguments

this_species	Species Temporal Data object. See sp_temporal_data for details.
train_prop	Numeric. Proportion of data to use for training. For example, a train_prop of 0.7 means 70% of the data will be used for training and 30% for testing.

Value

An object of class sp.temporal.env that consists in a list of five elements:

1. "temporal_df": a temporal data.frame (temporal_df) with the following columns: latitude, longitude, year, layer_dates, layers_path, cell_ids_year, and environmental data.
2. "sp_date_var": Name of date variable.
3. "lon_lat_vars": Names of the longitude and latitude variables.
4. "layers_ext": Environmental layers extension.
5. "env_data": Environmental data of occurrences.

Examples

```
library(tenm)
data("abronia")
tempora_layers_dir <- system.file("extdata/bio",package = "tenm")
abt <- tenm::sp_temporal_data(occs = abronia,
                              longitude = "decimalLongitude",
                              latitude = "decimalLatitude",
                              sp_date_var = "year",
                              occ_date_format="y",
                              layers_date_format= "y",
                              layers_by_date_dir = tempora_layers_dir,
                              layers_ext="*.tif$")
abtc <- tenm::clean_dup_by_date(abt,threshold = 10/60)
future::plan("multisession",workers=2)
abex <- tenm::ex_by_date(this_species = abtc,
                         train_prop=0.7)
future::plan("sequential")
```

---

inEllipsoid *inEllipsoid: Determine if a point is inside or outside an ellipsoid*

Description

Determine if a point is inside or outside an ellipsoid based on a confidence level.

Usage

```
inEllipsoid(centroid, eShape, env_data, level)
```

Arguments

centroid	Numeric vector of centroids for each environmental variable.
eShape	Shape matrix of the ellipsoid (can be a covariance matrix or a minimum volume ellipsoid).
env_data	Data frame with the environmental data.
level	Proportion of points to be included in the ellipsoids, equivalent to the error (E) proposed by Peterson et al. (2008).

Value

A data.frame with 2 columns:

- "in_Ellipsoid": Binary response indicating if each point is inside (1) or outside (0) the ellipsoid.
- "mh_dist": Mahalanobis distance from each point to the centroid.

Examples

```
library(tenm)
data("abronia")
tempora_layers_dir <- system.file("extdata/bio",package = "tenm")
abt <- tenm::sp_temporal_data(occs = abronia,
                              longitude = "decimalLongitude",
                              latitude = "decimalLatitude",
                              sp_date_var = "year",
                              occ_date_format="y",
                              layers_date_format= "y",
                              layers_by_date_dir = tempora_layers_dir,
                              layers_ext="*.tif$")
abtc <- tenm::clean_dup_by_date(abt,threshold = 10/60)
future::plan("multisession",workers=2)
abex <- tenm::ex_by_date(abtc,train_prop=0.7)
varcorrs <- tenm::correlation_finder(environmental_data = abex$env_data[,-ncol(abex$env_data)],
                                     method = "spearman",
                                     threshold = 0.8,
                                     verbose = FALSE)
future::plan("sequential")
mod <- tenm::cov_center(data = abex$env_data,
                        mve = TRUE,
                        level = 0.975,
                        vars = c("bio_05","bio_06","bio_12"))
in_elip <- tenm::inEllipsoid(centroid = mod$centroid,
                       eShape = mod$covariance,
                       env_data = abex$env_data[,c("bio_05","bio_06","bio_12")],
                       level = 0.975)
# 1 = Inside the ellipsoid; 0 = Outside the ellipsoid
print(in_elip)
```

metaras *Helper function to obtain layer name from a raster layer*

Description

Returns a character vector with the name of the raster layer.

Usage

```
metaras(r)
```

Arguments

r	An object of class SpatRaster representing the raster layer.

Value

A character vector with the name of the raster layer.

Examples

```
tempora_layers_dir <- system.file("extdata/bio",package = "tenm")
p1 <- list.files(tempora_layers_dir,full.names=TRUE,
                 pattern=".tif$",recursive=TRUE)[1]
r1 <- terra::rast(p1)
print(tenm::metaras(r1))
```

plot_ellipsoid *Function to plot ellipsoid models in E-space*

Description

The function plots 2D and 3D ellipsoids using environmental information as coordinates.

Usage

```
plot_ellipsoid(
  x,
  y,
  z = NULL,
  xlab = "x",
  ylab = "y",
  zlab = "x",
  mve = TRUE,
  level = 0.975,
  col = NULL,
```

```
  lwd_axes = 2,
  lty_axes = 2,
  semiaxes = FALSE,
  add = FALSE,
  ...
)
```

Arguments

x	Numeric vector representing the x coordinate of the ellipsoid.
y	Numeric vector representing the y coordinate of the ellipsoid.
z	Numeric vector representing the z coordinate of the ellipsoid. Defaults to NULL.
xlab	Character vector with the name of the x-axis label.
ylab	Character vector with the name of the y-axis label.
zlab	Character vector with the name of the z-axis label (if plotting in 3D).
mve	Logical. If TRUE, fits a minimum volume ellipsoid model.
level	Numeric value indicating the proportion of points to be included inside the ellipsoid model.
col	Plot color
lwd_axes	Line width for ellipsoid semi-axes.
lty_axes	Line type for ellipsoid semi-axes.
semiaxes	Logical. If TRUE, shows semi-axes of the ellipsoid.
add	Logical. If TRUE, add plot to existing plot (for 2D plots only).
...	Additional arguments to pass to base::plot, rgl::plot3d, rgl::wire3d, or other plotting functions

Value

A 2-dimensional or 3-dimensional plot depending on the input coordinates.

Examples

```
x <- rnorm(100)
y <- rnorm(100)
z <- rnorm(100)
# 2 dimensional plot
plot_ellipsoid(x, y, col = "darkgreen", xlab = "X-axis", ylab = "Y-axis",
               mve = TRUE, level = 0.95)
# 3 dimensional plot
plot_ellipsoid(x, y, z, col = "blue", xlab = "X-axis", ylab = "Y-axis",
               zlab = "Z-axis", mve = TRUE, level = 0.95)

# Examples using functions of the package
library(tenm)
data("abronia")
tempora_layers_dir <- system.file("extdata/bio",package = "tenm")
```

```
abt <- tenm::sp_temporal_data(occs = abronia,
                              longitude = "decimalLongitude",
                              latitude = "decimalLatitude",
                              sp_date_var = "year",
                              occ_date_format="y",
                              layers_date_format= "y",
                              layers_by_date_dir = tempora_layers_dir,
                              layers_ext="*.tif$")
abtc <- tenm::clean_dup_by_date(abt,threshold = 10/60)
future::plan("multisession",workers=2)
abex <- tenm::ex_by_date(abtc,train_prop=0.7)
future::plan("sequential")
x <- abex$temporal_df$bio_05
y <- abex$temporal_df$bio_06
z <- abex$temporal_df$bio_12
# 2D ellipsoid
tenm::plot_ellipsoid(x = x, y=y, semiaxes= TRUE,xlim=c(140,390))
tenm::plot_ellipsoid(x = x+100, y=y, semiaxes= TRUE,add=TRUE)
# 3D ellipsoid
tenm::plot_ellipsoid(x = x, y=y, z=z ,semiaxes= FALSE)
tenm::plot_ellipsoid(x = x+100, y=y, z=z ,semiaxes= FALSE,add=TRUE)
```

`predict,sp.temporal.selection-method`
Predict the potential distribution of species based on environmental conditions

---

Description

Predict the potential distribution of species based on environmental conditions

Usage

```
## S4 method for signature 'sp.temporal.selection'
predict(
  object,
  model_variables = NULL,
  layers = NULL,
  layers_path = NULL,
  layers_ext = NULL,
  mve = TRUE,
  level = 0.975,
  output = "suitability",
  ...
)
```

Arguments

object	An object of class sp.temporal.selection
model_variables	A character vector specifying the variable names used to build the model.
layers	A SpatRaster object or a list where each element is a SpatRaster.
layers_path	Path to the directory containing raster layers.
layers_ext	File extension of the raster layers.
mve	Logical indicating whether to use the minimum volume ellipsoid algorithm.
level	Proportion of data to include inside the ellipsoid if mve is TRUE.
output	Character indicating if the model outputs "suitability" values or "mahalanobis" distances.
...	Additional parameters passed to ellipsoid_projection.

Details

This function predicts the potential distribution of a species based on environmental conditions represented by raster layers. The prediction is based on the model statistics and environmental variables specified in 'model_variables'. If 'mve' is TRUE, the minimum volume ellipsoid algorithm is used to model the niche space. The output can be either "suitability", or "mahalanobis", indicating distance to the niche center. Note that each SpatRaster in the 'layers' parameter should have the same number of elements (layers) as 'model_variables'. The predict method assumes that variables in each SpatRaster correspond to those in 'model_variables'. If layers in the 'layers' parameter are given as a list of objects of class SpatRaster, then the number of prediction layers will have the same number of elements in the list.

Value

A SpatRaster object representing predicted suitability values or Mahalanobis distances to niche center.

Examples

```
library(tenm)
data("abronia")
tempora_layers_dir <- system.file("extdata/bio",package = "tenm")
abt <- tenm::sp_temporal_data(occs = abronia,
                              longitude = "decimalLongitude",
                              latitude = "decimalLatitude",
                              sp_date_var = "year",
                              occ_date_format="y",
                              layers_date_format= "y",
                              layers_by_date_dir = tempora_layers_dir,
                              layers_ext="*.tif$")
abtc <- tenm::clean_dup_by_date(abt,threshold = 10/60)
future::plan("multisession",workers=2)
abex <- tenm::ex_by_date(this_species = abtc,train_prop=0.7)
abbg <- tenm::bg_by_date(this_species = abex,
                         buffer_ngbs=NULL,n_bg=50000)
```

```
abbg <- tenm::bg_by_date(this_species = abex,
                         buffer_ngbs=10,n_bg=50000)
future::plan("sequential")
varcorrs <- tenm::correlation_finder(environmental_data =
                                       abex$env_data[,-ncol(abex$env_data)],
                                     method = "spearman",
                                     threshold = 0.8,
                                     verbose = FALSE)
mod_sel <- tenm::tenm_selection(this_species = abbg,
                                omr_criteria =0.1,
                                ellipsoid_level=0.975,
                                vars2fit = varcorrs$descriptors,
                                nvars_to_fit=c(3,4),
                                proc = TRUE,
                                RandomPercent = 50,
                                NoOfIteration=1000,
                                parallel=TRUE,
                                n_cores=2)
# Prediction using variables path
layers_70_00_dir <- system.file("extdata/bio_1970_2000",package = "tenm")
# The if the 'model_variables' parameter is set to NULL, the method uses
# the first model in the results table (mod_sel$mods_table)
suit_1970_2000 <- predict(mod_sel,
                          model_variables = NULL,
                          layers_path = layers_70_00_dir,
                          layers_ext = ".tif$")
# You can select the modeling variables used to project the model
suit_1970_2000 <- predict(mod_sel,
                          model_variables = c("bio_01","bio_04",
                                              "bio_07","bio_12"),
                          layers_path = layers_70_00_dir,
                          layers_ext = ".tif$")

# Pass a list containing the paths of the modeling layers
layers_1939_2016 <- file.path(tempora_layers_dir,c("1939","2016"))
suit_1939_2016 <- predict(mod_sel,model_variables = NULL,
                          layers_path = layers_1939_2016,
                          layers_ext = ".tif$")
# Pass a list of raster layers
layers_1939 <- terra::rast(list.files(layers_1939_2016[1],
                                      pattern = ".tif$",full.names = TRUE))
layers_2016 <- terra::rast(list.files(layers_1939_2016[2],
                                      pattern = ".tif$",full.names = TRUE))
layers_1939 <- layers_1939[[c("bio_01","bio_04","bio_07")]]
layers_2016 <- layers_2016[[c("bio_01","bio_04","bio_07")]]
layers_list <- list(layers_1939,layers_2016)
suit_1939_2016 <- predict(object = mod_sel,
                          model_variables = c("bio_01","bio_04","bio_07"),
                          layers_path = NULL,
                          layers = layers_list,
                          layers_ext = ".tif$")
```

pROC *Partial ROC calculation for Niche Models*

Description

Apply partial ROC tests to continuous niche models.

Usage

```
pROC(
  continuous_mod,
  test_data,
  n_iter = 1000,
  E_percent = 5,
  boost_percent = 50,
  rseed = FALSE,
  sub_sample = TRUE,
  sub_sample_size = 1000
)
```

Arguments

continuous_mod	A SpatRaster or numeric vector of the ecological niche model to be evaluated. If a numeric vector is provided, it should contain the values of the predicted suitability.
test_data	A numerical matrix, data.frame, or numeric vector: • If data.frame or matrix, it should contain coordinates of the occurrences used to test the ecological niche model. Columns must be: longitude and latitude. • If numeric vector, it should contain the values of the predicted suitability.
n_iter	Number of bootstrap iterations to perform for partial ROC calculations. Default is 1000.
E_percent	Numeric value from 0 to 100 used as the threshold (E) for partial ROC calculations. Default is 5.
boost_percent	Numeric value from 0 to 100 representing the percentage of testing data to use for bootstrap iterations in partial ROC. Default is 50.
rseed	Logical. Whether or not to set a random seed for reproducibility. Default is FALSE.
sub_sample	Logical. Indicates whether to use a subsample of the test data. Recommended for large datasets.
sub_sample_size	Size of the subsample to use for computing pROC values when sub_sample is TRUE.

Details

Partial ROC is calculated following Peterson et al. (2008; doi:10.1016/j.ecolmodel.2007.11.008). This function is a modification of the PartialROC function, available at https://github.com/narayanibarve/ENMGadgets.

Value

A list of two elements:

- "pROC_summary": a data.frame containing the mean AUC value, AUC ratio calculated for each iteration and the p-value of the test.
- "pROC_results": a data.frame with four columns containing the AUC (auc_model), partial AUC (auc_pmodel), partial AUC of the random model (auc_prand) and the AUC ratio (auc_ratio) for each iteration.

References

Peterson, A.T. et al. (2008) Rethinking receiver operating characteristic analysis applications in ecological niche modeling. Ecol. Modell., 213, 63–72. doi:10.1016/j.ecolmodel.2007.11.008

Examples

```
data(abronia)
suit_1970_2000 <- terra::rast(system.file("extdata/suit_1970_2000.tif",
                                          package = "tenm"))
print(suit_1970_2000)
proc_test <- tenm::pROC(continuous_mod = suit_1970_2000,
                        test_data = abronia[,c("decimalLongitude",
                                               "decimalLatitude")],
                        n_iter = 500, E_percent=5,
                        boost_percent=50)
print(proc_test$pROC_summary)
```

---

sp.temporal.bg-class *S3 classes to organize data and results of* tenm *objects*

Description

S3 classes to organize data and results of tenm objects

Value

An object of class 'sp.temporal.bg'. The object inherits information from objects of classes 'sp.temporal.modeling' and 'sp.temporal.env'. This class adds environmental background information.

Author(s)

Luis Osorio-Olvera

sp.temporal.env-class *S3 classes to organize data and results of* tenm *objects*

Description

S3 classes to organize data and results of tenm objects

Value

An object of class 'sp.temporal.env' inheriting information from 'sp.temporal.env'. This object adds a data.frame of environmental values associated with occurrence data.

Author(s)

Luis Osorio-Olvera

sp.temporal.modeling-class *S3 classes to organize data and results of* tenm *objects*

Description

S3 classes to organize data and results of tenm objects

Value

This object is a list comprising four elements: a) A data.frame containing occurrence records and layer information. b) A character vector specifying variable names. c) A character vector indicating the names of longitude and latitude variables. d) A character vector denoting the layers extension.

Author(s)

Luis Osorio-Olvera

sp.temporal.selection-class

S3 classes to organize data and results of `tenm` *objects*

Description

S3 classes to organize data and results of `tenm` objects

Value

An object of class 'sp.temporal.selection'. This object inherits information from objects of classes 'sp.temporal.modeling', 'sp.temporal.env' and 'sp.temporal.bg'. The object stores the results of the model calibration and selection process in a data.frame.

Author(s)

Luis Osorio-Olvera

sp_temporal_data

Function to create a Species Temporal Data object (STD object).

Description

Creates an object of class sp.temporal.modeling that contains a list with four attributes:

- temporal_df: A data frame with the following columns:
  - Longitude: Longitude coordinates of occurrence records.
  - Latitude: Latitude coordinates of occurrence records.
  - Date: Date variable indicating when the species were observed.
  - Layer Dates: Format of dates for each layer of environmental data.
  - Layers Path: Path to the bioclimatic layer corresponding to each year.
- sp_date_var: Name of the date variable column in the occurrence records.
- lon_lat_vars: Names of the longitude and latitude columns.
- layers_ext: Final extension format of the environmental information (e.g., ".tif").

Usage

```
sp_temporal_data(
  occs,
  longitude,
  latitude,
  sp_date_var,
  occ_date_format = "y",
  layers_date_format = "y",
  layers_by_date_dir,
  layers_ext = "*.tif$"
)
```

Arguments

occs	A data.frame with information about occurrence records of the species being modeled.
longitude	Name of the variable in 'occs' containing longitude data.
latitude	Name of the variable in 'occs' containing latitude data.
sp_date_var	Name of the date variable.
occ_date_format	Format of dates in occurrence records. Options: "y", "ym", "ymd", "mdy", "my", "dmy".
layers_date_format	Format of dates in raster layers. Options: "y", "ym", "ymd", "mdy", "my", "dmy".
layers_by_date_dir	Directory containing folders organized by date with raster layers of environmental information.
layers_ext	Extension or path of each raster layer archive (e.g., ".tif").

Details

The format of dates for each layer can be organized in a particular pattern, for example year/month/day ("ymd"), year/month ("ym"), just year ("y") or some other arrangement like month/year ("my"), month/year/day ("myd"), day/month/year ("dmy").

Value

Returns a sp.temporal.modeling object (list) with the coordinates of each occurrences points, the years of observation and the path to the temporal layers.

Examples

```
library(tenm)
#A data.frame with occurrences points information of Abronia graminea.
# See help(abronia)
data("abronia")
tempora_layers_dir <- system.file("extdata/bio",package = "tenm")
abt <- tenm::sp_temporal_data(occs = abronia,
                              longitude = "decimalLongitude",
                              latitude = "decimalLatitude",
                              sp_date_var = "year",
                              occ_date_format="y",
                              layers_date_format= "y",
                              layers_by_date_dir = tempora_layers_dir,
                              layers_ext="*.tif$")
```

tdf2swd *Temporal data.frame to Samples With Data format*

Description

Converts a temporal data.frame to Samples With Data (SWD) table for use with other modeling platforms such as MaxEnt.

Usage

```
tdf2swd(this_species, sp_name = "sp")
```

Arguments

this_species	An object of class sp.temporal.env (see ex_by_date) or sp.temporal.bg (see bg_by_date).
sp_name	Character vector specifying the species name.

Value

A data.frame formatted as Samples With Data (SWD) table.

Examples

```
library(tenm)
data("abronia")
tempora_layers_dir <- system.file("extdata/bio",package = "tenm")
abt <- tenm::sp_temporal_data(occs = abronia,
                              longitude = "decimalLongitude",
                              latitude = "decimalLatitude",
                              sp_date_var = "year",
                              occ_date_format="y",
                              layers_date_format= "y",
                              layers_by_date_dir = tempora_layers_dir,
                              layers_ext="*.tif$")
abtc <- tenm::clean_dup_by_date(abt,threshold = 10/60)
future::plan("multisession",workers=2)
abex <- tenm::ex_by_date(this_species = abtc,
                         train_prop=0.7)
abbg <- tenm::bg_by_date(abex,
                         buffer_ngbs=10,n_bg=50000)
future::plan("sequential")
# SWD table for occurrence records
occ_swd <- tdf2swd(this_species=abex,sp_name="abro_gram")
# SWD table for background data
bg_swd <- tdf2swd(this_species=abbg)
```

tenm_selection *Function to find the best n-dimensional ellipsoid model*

Description

Finds the best n-dimensional ellipsoid model using a model calibration and selection protocol for ellipsoid models.

Usage

```
tenm_selection(
  this_species,
  omr_criteria = 0.1,
  ellipsoid_level = 0.975,
  vars2fit,
  nvars_to_fit = c(2, 3),
  mve = TRUE,
  proc = TRUE,
  sub_sample = TRUE,
  sub_sample_size = 1000,
  RandomPercent = 50,
  NoOfIteration = 1000,
  parallel = TRUE,
  n_cores = 4
)
```

Arguments

this_species	An object of class sp.temporal.env representing species occurrence data organized by date. See ex_by_date.
omr_criteria	Omission rate criterion used to select the best models. See ellipsoid_selection for details
ellipsoid_level	Proportion of points to include inside the ellipsoid.
vars2fit	A vector of variable names to use in building the models.
nvars_to_fit	Number of variables used to build the models.
mve	Logical. If TRUE, a minimum volume ellipsoid will be computed.
proc	Logical. If TRUE, compute the partial ROC test for each model.
sub_sample	Logical. Indicates whether to use a subsample of size sub_sample_size for computing pROC values, recommended for large datasets.
sub_sample_size	Size of the sub_sample to use for computing pROC values when sub_sample is TRUE.
RandomPercent	Percentage of occurrence points to sample randomly for bootstrap in the Partial ROC test. See pROC.

NoOfIteration	Number of iterations for the bootstrap in the Partial ROC test. See pROC.
parallel	Logical. Whether to run computations in parallel. Default is TRUE.
n_cores	Number of cores to use for parallelization. Default is 4.

Value

An object of class "sp.temporal.selection" containing metadata of model statistics of the calibrated models, obtainable from the "mods_table" attribute. The function internally uses ellipsoid_selection to obtain model statistics. Note that this function inherits attributes from classes "sp.temporal.modeling" (see sp_temporal_data), "sp.temporal.env" (see ex_by_date), and "sp.temporal.bg" (see bg_by_date), thus all information from these classes can be extracted from this object.

Examples

```
library(tenm)
data("abronia")
tempora_layers_dir <- system.file("extdata/bio",package = "tenm")
abt <- tenm::sp_temporal_data(occs = abronia,
                              longitude = "decimalLongitude",
                              latitude = "decimalLatitude",
                              sp_date_var = "year",
                              occ_date_format="y",
                              layers_date_format= "y",
                              layers_by_date_dir = tempora_layers_dir,
                              layers_ext="*.tif$")
abtc <- tenm::clean_dup_by_date(abt,threshold = 10/60)
future::plan("multisession",workers=2)
abex <- tenm::ex_by_date(this_species = abtc,
                         train_prop=0.7)
abbg <- tenm::bg_by_date(abex,
                         buffer_ngbs=10,n_bg=50000)
future::plan("sequential")
varcorrs <- tenm::correlation_finder(environmental_data = abex$env_data[,-ncol(abex$env_data)],
                                     method = "spearman",
                                     threshold = 0.8,
                                     verbose = FALSE)
vars2fit <- varcorrs$descriptors
mod_sel <- tenm::tenm_selection(this_species = abbg,
                                omr_criteria =0.1,
                                ellipsoid_level=0.975,
                                vars2fit = vars2fit,
                                nvars_to_fit=c(2,3),
                                proc = TRUE,
                                RandomPercent = 50,
                                NoOfIteration=1000,
                                parallel=TRUE,
                                n_cores=20)
# Project potential distribution using bioclimatic layers for 1970-2000
# period.
layers_70_00_dir <- system.file("extdata/bio_1970_2000",package = "tenm")
suit_1970_2000 <- predict(mod_sel,model_variables = NULL,
                          layers_path = layers_70_00_dir,
```

```
                          layers_ext = ".tif$")
terra::plot(suit_1970_2000)
colors <- c('#000004FF', '#040312FF', '#0B0725FF',
            '#0B0725FF', '#160B38FF', '#160B38FF',
            '#230C4CFF', '#310A5CFF', '#3F0966FF',
            '#4D0D6CFF', '#5A116EFF', '#67166EFF',
            '#741A6EFF', '#81206CFF', '#81206CFF',
            '#8E2469FF', '#9B2964FF', '#A82E5FFF',
            '#B53359FF', '#B53359FF', '#C03A50FF',
            '#CC4248FF', '#D74B3FFF', '#E05536FF',
            '#E9602CFF', '#EF6E21FF', '#F57B17FF',
            '#F8890CFF', '#FB9806FF', '#FB9806FF',
            '#FCA70DFF', '#FBB81DFF', '#F9C72FFF',
            '#F9C72FFF', '#F6D847FF', '#F2E763FF',
            '#F2E763FF', '#F3F585FF', '#FCFFA4FF',
            '#FCFFA4FF')
points(abtc$temporal_df[,1:2],pch=17,cex=1,
       col=rev(colors))
legend("topleft",legend = abtc$temporal_df$year[1:18],
       col =rev(colors[1:18]),
       cex=0.75,pch=17)
legend("topright",legend = unique(abtc$temporal_df$year[19:40]),
       col = rev(colors[19:40]),
       cex=0.75,pch=17)
```

Index

∗ **datasets**
abronia, 2
colors, 9

abronia, 2

bg_by_date, 3, *31*, *33*

cells2samp, 4
clean_dup, 6, *8*
clean_dup_by_date, 8
colors, 9
correlation_finder, 10, *16*
cov.mve, *11*
cov.rob, *13*, *16*
cov_center, 11, *14*

ellipsoid_omr, 12, *17*
ellipsoid_projection, 14, *24*
ellipsoid_selection, 16, *32, 33*
ex_by_date, *3*, 18, *31–33*

inEllipsoid, *17*, 19

metaras, 21

plot3d, *15*
plot_ellipsoid, 21
predict
(predict,sp.temporal.selection-method), 23
predict,sp.temporal.selection-method, 23
pROC, 26, *32, 33*

sp.temporal.bg-class, 27
sp.temporal.env-class, 28
sp.temporal.modeling-class, 28
sp.temporal.selection-class, 29
sp_temporal_data, *8*, *19*, 29, *33*

tdf2swd, 31
tenm_selection, *4*, 32