Package ‘dynamicSDM’

December 6, 2022

Title Dynamic species distribution and abundance modelling

Version 1.0

Description This package enables the generation of species distribution and abundance models that are dynamic through space and time. dynamicSDM presents highly-flexible functions for incorporating both spatial and temporal aspects at key species distribution modelling stages, including when cleaning and filtering species occurrence data, assessing and correcting biases and autocorrelation, generating pseudo-absence records, extracting dynamic explanatory variables and projecting species distribution and abundance patterns at high spatiotemporal resolution. Functions take advantage of Google Earth Engine and Google Drive to minimise computing power and storage demands.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Imports ape, dplyr, gbm, gargle, geodist, geosphere, ggplot2, googledrive, lubridate, magick, magrittr, matrixStats, rangemap, raster, reticulate, rgee, rgeos, sp, stats, stringr, terra, tidyr, grDevices, graphics, methods, utils

Depends R (>= 3.5.0)

LazyData true

Suggests covr, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Rachel Dobson [aut, cre] (<https://orcid.org/0000-0003-3990-267X>), Andy J. Challinor [aut] (<https://orcid.org/0000-0002-8551-6617>), Robert A. Cheke [aut] (<https://orcid.org/0000-0002-7437-1934>), Stewart Jennings [aut] (<https://orcid.org/0000-0002-1267-8623>), Stephen G. Willis [aut] (<https://orcid.org/0000-0002-8656-5808>), Martin Dallimer [aut] (<https://orcid.org/0000-0001-8120-3309>)

Maintainer Rachel Dobson <eerdord@gmail.com>

R topics documented:

`biome_layer` *Categorical raster.*

Description

Categorical RasterLayer cropped to southern Africa at 10 degree resolution. Represents a biome classification RasterLayer that could be used when spatially blocking species occurrence data.

Usage

```
biome_layer
```

Format

A RasterLayer

class RasterLayer

dimensions 4, 3, 12 (nrow, ncol, ncell)

resolution 10, 10 (x, y)

extent 11.71845, 40.85081, -47.89832, -4.428839 (xmin, xmax, ymin, ymax)

crs "+proj=longlat +datum=WGS84 +no_defs"

source memory

names layer

values Numerical categorical value.

---

brt_fit   *Fit boosted regression tree models to species distribution or abundance data.*

Description

Fit gradient boosting boosted regression tree models to species distribution and abundance data and associated dynamic explanatory variables.

Usage

```
brt_fit(
  occ.data,
  response.col,
  varnames,
  distribution,
  block.col = NULL,
  weights.col = NULL,
  test.data = NULL,
  interaction.depth = NULL,
  n.trees = 5000,
  shrinkage = 0.001
)
```

Arguments

`occ.data`	a data frame, the data to fit boosted regression tree models to, containing columns for model response and explanatory variable data. If required, occ.data should contain block.col and weights.col columns too.
`response.col`	a character string, the name of the column in occ.data containing response variable column.
`varnames`	a character vector, the names of the columns containing model explanatory variables in occ.data.
`distribution`	a character string, the model distribution family to use, such as "Gaussian", "Poisson" or "Bernoulli".
`block.col`	optional; a character string, the name of the column in occ.data containing spatiotemporal block numbers for occ.data splitting. See details for more information.
`weights.col`	a character string, the name of the column in occ.data containing spatiotemporal sampling effort weights to be used in fitting process.
`test.data`	optional; a data frame, the testing dataset for optimising interaction.depth when blocking is not used.
`interaction.depth`	optional; an integer specifying the maximum depth of each tree (i.e. highest level of variable interactions allowed). Default optimises depth between 1 and 4.

`n.trees` optional; an integer, the number of trees in boosted regression tree models. Default is 5000.

`shrinkage` optional; an integer, the shrinkage parameter applied to each tree in the boosted regression tree expansion. Also known as learning rate. Default is 0.001.

Details

This function calculates a gradient boosting “gbm” object for the response and explanatory variable data provided, using the gbm R package (Greenwell et al., 2019). dynamicSDM adds extra functionality for optimising the gbm interation.depth parameter and splitting training and testing data by spatiotemporal blocks to account for spatial and temporal autocorrelation.

If interaction.depth is not given, then brt_fit will vary the interaction.depth parameter between 1 (an additive model) and 4 (four-way interaction model). For each interaction.depth value, model performance is measured by calculating the root-mean-square error of model predictions compared to actual values in the testing data. The interaction.depth value that results in the lowest root-mean-square error is used for the returned fitted model.

The model testing dataset can either be given using argument test.data or block.col given. In the latter situation, each unique block.col block is excluded in a jack-knife approach following Bagchi et al., (2013). This approach uses each block as the model testing dataset in numerical order, whilst all other block.col blocks are used as training data for the boosted regression tree model. Therefore, the function returns a list of fitted boosted regression tree models equal to the length of unique blocking categories in block.col. If block.col is not given, models are fit to all occ.data and a single gbm model is returned.

If weights.col given, records are weighted by their associated value in the weights.col when model fitting. For instance, the user may wish to down weigh the importance of records collected at oversampled sites and times when fitting models, and vice versa, to account for spatiotemporal biases in occurrence records.

Value

Returns a “gbm” model object or list of “gbm” model objects.

References

Bagchi, R., Crosby, M., Huntley, B., Hole, D. G., Butchart, S. H. M., Collingham, Y., Kalra, M., Rajkumar, J., Rahmani, A. & Pandey, M. 2013. Evaluating the effectiveness of conservation site networks under climate change: accounting for uncertainty. Global Change Biology, 19, 1236-1248. Greenwell, B., Boehmke, B., Cunningham, J., & GBM Developers. 2019. Package ‘gbm’. R package version, 2.

Examples

```
data("sample_model_data")
split <- sample(c(TRUE, FALSE),
               replace=TRUE,
               nrow(sample_model_data),
               prob = c(0.75, 0.25))
training <- sample_model_data[split, ]
testing <- sample_model_data[!split, ]
brt_fit(
 occ.data = training,
 test.data = testing,
 response.col = "presence.absence",
```

```
 distribution = "bernoulli",
 weights.col = "sampling_weights",
 varnames = colnames(training)[9:12]
)

training <- sample_model_data
brt_fit(
 occ.data = training,
 response.col = "presence.absence",
 distribution = "bernoulli",
 block.col = "blockno",
 weights.col = "sampling_weights",
 varnames = colnames(training)[9:12]
)
```

`count` *count computes the number of non-null inputs.*

Description

count computes the number of non-null inputs.

Usage

```
count(x, na.omit = T)
```

`dynamic_proj` *Project species distribution and abundance models onto dynamic environmental covariates.*

Description

Projects fitted species distribution and abundance models onto projection covariates for each date given.

Usage

```
dynamic_proj(
  dates,
  projection.method,
  local.directory = NULL,
  drive.folder = NULL,
  user.email = NULL,
  sdm.mod = NULL,
  sdm.thresh = 0.5,
  sdm.weight = 1,
  sam.mod = NULL,
  sam.weight = 1,
  save.directory = NULL,
  save.drive.folder = NULL
)
```

Arguments

`dates` a character string, vector of dates in format YYYY-MM-DD.

`projection.method` a character string or vector, the method or methods to project distribution and abundance onto projection covariates. Options include "'proportional'", "'binary'", "'abundance'" and "'stacked'". See details for more information.

`local.directory` optional; a character string, the path to a local directory to read projection covariate data frames from.

`drive.folder` optional; a character string, the Google Drive folder to read projection covariate data frames from.

`user.email` optional; a character string, user email for initialising Google Drive. Required if drive.folder or save.drive.folder used.

`sdm.mod` optional; a model object or list of model objects fitted to species distribution data.

`sdm.thresh` optional; a numeric value, the threshold to convert projected distribution suitability into binary presence-absence. Default 0.5. Required if projection.method is "'binary'" or "stacked".

`sdm.weight` optional; a numeric string, weights given to each sdm.mod model projection, given in the same order as the sdm.mod list. Default is equal weighting to all models.

`sam.mod` optional; a model object or list of model objects fitted to species abundance data.

`sam.weight` optional; a numeric string, weights given to each sdm.mod model projection, given in the same order as the sam.mod list. Default is equal weighting to all models.

`save.directory` optional; a character string, path to local directory to save projection rasters to.

`save.drive.folder` optional; a character string, Google Drive folder to save projection rasters to.

Details

Function projects a model object or list of model objects onto projection covariate data frames for each projection date given.

Projection covariate data frames must be saved ".csv" files in the drive.folder or save.drive.folder given, and must be unique in containing the relevant projection date in "YYYY-MM-DD" format. For instance, two or more ".csv" files saved with the given Google Drive folder or local directory that contain the projection date will result in function error. Additionally, column names of projection covariate data frames must match the explanatory variable names that fitted models are trained on.

When multiple models are provided, the function projects each model onto the projection covariates and takes the average value across all model projections. However, if users have specified sdm.weight or sam.weight then the weighted average of model projections is returned. For example, this could be used to down weight projections of poorly performing models in an ensemble.

The "'proportional'" projection.method projects sdm.mod model objects onto projection covariates for each date, exporting rasters for projected distribution suitability, a continuous measure between 0 (least suitable) and 1 (most suitable).

The "'binary'" projection.method projects sdm.mod onto projection covariates for each date, exporting rasters for projected binary presence (1) or absence (0), derived from "'proportional'" projected distribution suitability using user-specified threshold (sdm.thresh) or default threshold of 0.5.

The "'abundance'" projection.method projects sam.mod onto projections covariates for each date, exporting rasters for projected abundance in the abundance units that sam.mod were fitted onto.

The "stacked" projection.method follows the "'binary'" projection.method and then projects abundance onto only binary presence (1) cells using "abundance" projection.method.

One or both of save.drive.folder and save.directory are required to specify where projection rasters are to be saved.

If one of drive.folder or save.drive.folder are used then user.email is required to access the appropriate Google Drive user account. This requires users to have installed R package "googledrive" and initialised Google Drive with valid log-in credentials. Please follow instructions on https://googledrive.tidyverse.org/.

Value

Exports model projection rasters for each projection date to user-specified Google Drive folder or local directory.

---

dynamic_proj_covariates

Combine explanatory variable rasters into a covariate data frame for each projection date.

Description

Explanatory variable rasters are imported for each projection date, resampled to given spatial resolution and extent and stacked, and then written to a covariate data frame for each projection date.

Usage

```
dynamic_proj_covariates(
  dates,
  varnames,
  drive.folder = NULL,
  user.email = NULL,
  local.directory = NULL,
  spatial.ext = NULL,
  spatial.res.degrees = NULL,
  resample.method = NULL,
  save.directory = NULL,
  save.drive.folder = NULL
)
```

Arguments

`dates`	a character string, vector of dates in format YYYY-MM-DD.
`varnames`	a character string, the unique names for each explanatory variable.
`drive.folder`	optional; a character string or vector, Google Drive folder or folders to read projection covariate rasters from.
`user.email`	optional; a character string, user email for initialising Google Drive. Required if drive.folder or save.drive.folder used.

`local.directory`
: optional; a character string or vector, path to local directory or directories to read projection covariate rasters from.

`spatial.ext`
: optional; the spatial extent to crop explanatory variable rasters to. Object of class "Extent", "RasterLayer" or "polygon" or numeric vector listing xmin, xmax, ymin and ymax in order.

`spatial.res.degrees`
: optional; a numeric value, the spatial resolution in degrees for projection rasters to be resampled to. Required if spatial.ext given.

`resample.method`
: a character string or vector length of varnames, specifying resampling method to use. One of "ngb" and "bilinear". See details for more information.

`save.directory`
: optional; a character string, path to local directory to save projection covariate data frames to.

`save.drive.folder`
: optional; a character string, Google Drive folder to save projection covariate data frames to.

Details

For each projection date, appropriate rasters for each explanatory variable are imported from a local directory or Google Drive folder. If required, rasters are cropped and resampled to the same spatial extent and resolution. Rasters are then stacked and transformed into a covariate data frame with column names matching the unique explanatory variable names. A data frame of these projection covariates are exported to a local directory or Google Drive folder as ".csv" files named as the relevant date in "YYYY-MM-DD" format. Note: if explanatory variable rasters are not of the same spatial resolution and extent, then the function will error. Resample methods (resample.method) include: "ngb", in which each cell acquires the value of its nearest neighbour cell in the original raster and is often used for categorical variables; and "bilinear", in which the distance-weighted average of the four nearest cells are used to estimate a new cell value and is often used for continuous variables. If only one resample.method is given, but these are more than one explanatory variables, the same resample.method is used for all.

If drive.folder or save.drive.folder arguments are used to download rasters for use or upload function output, then users must have installed R package "googledrive" and initialised Google Drive with valid log-in credentials. The credentials must be given under function argument user.email to initiate the correct Google Drive account. Please follow instructions on https://googledrive.tidyverse.org/.

`dynamic_proj_dates` *Generate vector of dates for dynamic projections*

Description

Creates a vector of dates at regular intervals between two given dates.

Usage

```
dynamic_proj_dates(startdate, enddate, interval.level, interval)
```

Arguments

startdate	a character string, the start date in format YYYY-MM-DD.
enddate	a character string, the end date in format YYYY-MM-DD.
interval.level	a character string, the time-step of intervals. One of '"day"', '"month"' or '"year"': can be abbreviated.
interval	a numeric value, the length of interval in interval.level units to generate between the start and end date.

Details

Function returns a vector of dates given start.date and end.date at given interval size.

Value

Vector of dates between start date and end date split at regular intervals.

Examples

```
dynamic_proj_dates(
 startdate = "2000-01-01",
 enddate = "2001-01-01",
 interval.level = "month",
 interval = 2
)
```

---

dynamic_proj_GIF *Create GIF of dynamic species distribution and abundance projections*

Description

Plots dynamic species distribution and abundance projections through time and combines images into a GIF.

Usage

```
dynamic_proj_GIF(
  dates,
  projection.type,
  drive.folder = NULL,
  user.email = NULL,
  local.directory = NULL,
  save.drive.folder = NULL,
  save.directory = NULL,
  width = 480,
  height = 480,
  legend.max = NULL,
  legend.min = NULL,
  legend.name = NULL,
  file.name = NULL
)
```

Arguments

- `dates` — a character vector , projection dates in format YYYY-MM-DD.
- `projection.type` — a character string, the type of distribution or abundance projection to plot. One of "'proportional'", "'binary'", "'abundance'" and "'stacked'".
- `drive.folder` — optional; a character string, the Google Drive folder to read projection rasters from.
- `user.email` — optional; a character string, user email for initialising Google Drive. Required if drive.folder or save.drive.folder used.
- `local.directory` — optional; a character string, the path to local directory to read projection rasters from.
- `save.drive.folder` — optional; a character string, Google Drive folder to save GIF to.
- `save.directory` — optional; a character string, path to local directory to save GIF to.
- `width` — optional; a numeric value, the GIF width in pixels. Default = 480.
- `height` — optional; a numeric value, the GIF height in pixels. Default = 480.
- `legend.max` — optional; a numeric value, the maximum limit of legend values to standardise across projections.
- `legend.min` — optional; a numeric value, the minimum limit of legend values to standardise across projections.
- `legend.name` — optional; a character string, the name for the legend title. Default = projection.type.
- `file.name` — optional, a character string, the name for the saved GIF file. Default = projection.type.

Details

Function reads in projection rasters for each date and projection.type. These are plotted using ggplot2 and combined into Graphics Interchange Format (GIF).

For dynamic_proj_GIF to find the projection rasters for each date, then ".tif" files must be uniquely named with the date in format YYYY-MM-DD and projection.type. If more than one file name matches the date and projection.type, the function will error.

If one of drive.folder or save.drive.folder is used then user.email for the Google Drive account must be provided. This requires users to have installed R package "googledrive" and initialised Google Drive with valid log-in credentials. Please follow instructions on https://googledrive.tidyverse.org/.

Value

Exports GIF to Google Drive folder or local directory.

References

Wickham, H., and Chang, W, 2016. Package 'ggplot2'. Create elegant data visualisations using the grammar of graphics. Version, 2(1), pp.1-189.

`extract_buffered_coords`
Extract spatially buffered and temporally dynamic explanatory variable data for occurrence records.

Description

For each species occurrence record co-ordinate and date, spatially buffered and temporally dynamic explanatory data are extracted using Google Earth Engine.

Usage

```
extract_buffered_coords(
  occ.data,
  datasetname,
  bandname,
  spatial.res.metres,
  GEE.math.fun,
  moving.window.matrix,
  extraction.drive.folder,
  user.email,
  save.method,
  varname = NULL,
  temporal.res = NULL,
  temporal.level = NULL,
  temporal.direction = NULL,
  categories = NULL,
  save.directory = NULL
)
```

Arguments

`occ.data`
: a data frame, with columns for occurrence record co-ordinates and dates with column names as follows; record longitude as "x", latitude as "y", year as "year", month as "month", and day as "day".

`datasetname`
: a character string, the Google Earth Engine dataset to extract data from.

`bandname`
: a character string, the Google Earth Engine dataset bandname to extract data for.

`spatial.res.metres`
: a numeric value, the spatial resolution in metres for data extraction.

`GEE.math.fun`
: a character string, the mathematical function to compute across the specified spatial matrix and period for each record.

`moving.window.matrix`
: a matrix of weights with an odd number of sides, representing the spatial neighbourhood of cells ("moving window") to calculate GEE.math.fun across from record co-ordinate. See details for more information.

`extraction.drive.folder`
: a character string, Google Drive folder to temporarily save rasters to using Google Earth Engine before spatial buffering in R.

`user.email`
: a character string, user email for initialising Google Drive.

save.method — a character string, the method used to save extracted variable data. One of '"split"' or '"combined"': can be abbreviated. See details.

varname — optional; a character string, a unique name for the explanatory variable. Default varname is "bandname_temporal.res_temporal.direction_ GEE.math.fun_buffered".

temporal.res — optional; a numeric value, the temporal resolution in days to extract data and calculate GEE.math.fun across from occurrence record date.

temporal.level — a character string, the temporal resolution of the explanatory variable data. One of '"day"', '"month"' or '"year"': can be abbreviated. Default; day.

temporal.direction — optional; a character string, the temporal direction for extracting data across relative to the record date. One of '"prior"' or '"post"': can be abbreviated.

categories — optional; a character string, the categories to use in calculation if data are categorical. See details for more information.

save.directory — a character string, path to a local directory to save extracted variable data to.

Details

For each individual species occurrence record co-ordinate and date, this function extracts data for a given band within a Google Earth Engine dataset across a user-specified spatial buffer and temporal period and calculates a mathematical function on such data.

If temporal.res and temporal.direction are not given, the function extracts explanatory variable data (in RasterLayer format) for all of the cells surrounding and including the cell containing the occurrence record co-ordinates. If temporal.res and temporal.direction is given, the function extracts explanatory variable data (in RasterLayer format) for which GEE.math.fun has been first calculated over this period in relation to the occurrence record date.

Using the focal function in raster R package (Hijmans et al., 2015), GEE.math.fun is calculated across the spatial buffer area from the record co-ordinate. The spatial buffer area used is specified by argument moving.window matrix, which dictates the neighbourhood of cells surrounding the cell containing the occurrence record to include in the calculation.

When explanatory variable data are categorical (e.g. land cover classes), argument categories can be used to specify the categories of importance to the calculation. The category or categories given will be converted in a binary representation, with "1" for those listed, and "0" for all others in the dataset. Ensure that the GEE.math.fun given is appropriate for such data. For example, this function could return the sum of suitable land cover classified cells in the "moving window" from species occurrence record co-ordinates.

extract_buffered_coords requires users to have installed R package "rgee" (Aybar et al., 2020) and initialised Google Earth Engine with valid log-in credentials. Please follow instructions on the following website https://cran.r-project.org/web/packages/rgee/vignettes/rgee01.html. datasetname must be in the accepted Google Earth Engine Data catalogue layout (e.g. "MODIS/006/MCD12Q1" or "UCSB-CHG/CHIRPS/DAILY") and bandname as specified in the dataset (e.g. "LC_Type5", "precipitation"). For datasets and band names, see https://developers.google.com/earth-engine/datasets.

extract_buffered_coords also requires users to have installed the R package "googledrive" (D'Agostino McGowan and Bryan, 2022) and initialised Google Drive with valid log-in credentials, which must be provided under argument user.email. Please follow instructions on https://googledrive.tidyverse.org/ for initialising the googledrive package.

GEE.math.fun specifies the mathematical function to be calculated over the spatial buffered area and temporal period. Options are limited to Google Earth Engine ImageCollection Reducer functions (https://developers.google.com/earth-engine/apidocs/) for which an analogous R function is available. This includes: "allNonZero","anyNonZero", "count", "first","firstNonNull", "last", "lastNonNull", "max","mean", "median","min", "mode","product", "sampleStdDev", "sampleVariance", "stdDev", "sum" and "variance".

temporal.level states the temporal resolution of the explanatory variable data and improves the speed of extract_buffered_coords extraction. For example, if the explanatory data represents an annual variable, then all record co-ordinates from the same year can be extracted from the same raster. However, if the explanatory data represents a daily variable, then only records from the exact same day can be extracted from the same raster. For the former, temporal.level argument should be 'year' and for the latter, temporal.level should be 'day'.

For save.method '"combined"', the function with save ".csv" files containing all occurrence records and associated values for the explanatory variable. If save.method '"split"' is chosen, the function will save individual ".csv" files for all of the records with each unique period of the given temporal.level (e.g. each year, each year and month combination or each unique date). '"split"' method is provided to protect users if internet connection is lost when extracting data for large occurrence datasets.

Value

Returns details of successful explanatory variable extractions.

References

Aybar, C., Wu, Q., Bautista, L., Yali, R. and Barja, A., 2020. rgee: An R package for interacting with Google Earth Engine. Journal of Open Source Software, 5(51), p.2272.

D'Agostino McGowan L., and Bryan J., 2022. googledrive: An Interface to Google Drive. https://googledrive.tidyverse.o https://github.com/tidyverse/googledrive.

Hijmans, R. J., Van Etten, J., Cheng, J., Mattiuzzi, M., Sumner, M., Greenberg, J. A., Lamigueiro, O. P., Bevan, A., Racine, E. B. & Shortridge, A. 2015. Package 'raster'. R package, 734.

`extract_buffered_raster` *Extract spatially buffered and temporally dynamic rasters of explanatory variable data.*

Description

Extract rasters for spatially buffered and temporally dynamic explanatory variables at each projection date using Google Earth Engine.

Usage

```
extract_buffered_raster(
  dates,
  spatial.ext,
  datasetname,
  bandname,
  spatial.res.metres,
  GEE.math.fun,
  moving.window.matrix,
  user.email,
```

```
  varname = NULL,
  temporal.res = NULL,
  temporal.direction = NULL,
  categories = NULL,
  save.directory = NULL,
  save.drive.folder
)
```

Arguments

`dates` a character string, vector of dates in format YYYY-MM-DD.

`spatial.ext` the spatial extent for the extracted raster. Object of class "Extent", "RasterLayer" or "polygon" or numeric vector listing xmin, xmax, ymin and ymax in order.

`datasetname` a character string, the Google Earth Engine dataset to extract data from.

`bandname` a character string, the Google Earth Engine dataset bandname to extract data for.

`spatial.res.metres` a numeric value, specifying the spatial resolution in metres of the raster to be extracted.

`GEE.math.fun` a character string, the mathematical function to compute across the specified period and spatial buffer from each projection date and cell.

`moving.window.matrix` “moving window” matrix of weights with an odd number of sides to specify spatial neighbourhood of cells to calculate GEE.math.fun across for each cell in spatial.ext. See details for more information.

`user.email` a character string, user email for initialising Google Drive.

`varname` optional; a character string, the unique name for the explanatory variable. Default varname is “bandname_temporal.res_temporal.direction_ GEE.math.fun_buffered_raster".

`temporal.res` optional; a numeric value, the temporal resolution in days prior or post each projection date to calculate GEE.math.fun across.

`temporal.direction` optional; a character string, the temporal direction for extracting dynamic variable data across relative to each projection date given. One of ’"prior"’ or ’"post"’: can be abbreviated.

`categories` optional; a character string, the categories to use in the calculation if data are categorical. See details for more information.

`save.directory` optional; a character string, path to local directory to save extracted rasters to.

`save.drive.folder` a character string, Google Drive folder to save extracted rasters to.

Details

For each projection date, this function downloads rasters at given spatial extent and resolution for spatially buffered and temporally dynamic explanatory variables. Rasters are saved directly to Google Drive, with option to export to local directory too. These can be combined to create projection covariate data frames for projection dynamic species distribution and abundance at high spatiotemporal resolution

If temporal.res and temporal.direction are not given, the function extracts explanatory variable data (in RasterLayer format) for all of the cells in spatial extent (spatial.ext) given. If temporal.res and temporal.direction is given, the function extracts explanatory variable data (in RasterLayer format) for which GEE.math.fun has been first calculated over this period in relation to the projection date.

Using the focal function in raster R package (Hijmans et al., 2015), GEE.math.fun is calculated across the spatial buffer area from each cell across spatial extent given. The spatial buffer area used is specified by argument moving.window matrix, which dictates the neighbourhood of cells surrounding each cell in spatial.ext to include in the calculation.

When explanatory variable data are categorical (e.g. land cover classes), argument categories can be used to specify the categories of importance to the calculation. The category or categories given will be converted in a binary representation, with “1” for those listed, and “0” for all others in the dataset. Ensure that the GEE.math.fun given is appropriate for such data. For example, this function could return the sum of suitable land cover classified cells in the “moving window” from each cell across spatial extent given.

extract_buffered_raster requires users to have installed R package "rgee" (Aybar et al., 2020) and initialised Google Earth Engine with valid log-in credentials. Please follow instructions on the following website . datasetname must be in the accepted Google Earth Engine Data catalogue layout (e.g. “MODIS/006/MCD12Q1” or “UCSB-CHG/CHIRPS/DAILY”) and bandname as specified in the dataset (e.g. “LC_Type5”, “precipitation”). For datasets and band names, see .

extract_buffered_raster also requires users to have installed the R package "googledrive" (D’Agostino McGowan and Bryan, 2022) and initialised Google Drive with valid log-in credentials, which must be provided under argument user.email. Please follow instructions on https://googledrive.tidyverse.org/ for initialising the googledrive package.

GEE.math.fun specifies the mathematical function to be calculated over the spatial buffered area and temporal period. Options are limited to Google Earth Engine ImageCollection Reducer functions (https://developers.google.com/earth-engine/apidocs/) for which an analogous R function is available. This includes: "allNonZero","anyNonZero", "count", "first","firstNonNull", "last", "lastNonNull", "max","mean", "median","min", "mode","product", "sampleStdDev", "sampleVariance", "stdDev", "sum" and "variance".

Value

Returns details of successful explanatory variable extractions for each projection date.

References

Aybar, C., Wu, Q., Bautista, L., Yali, R. and Barja, A., 2020. rgee: An R package for interacting with Google Earth Engine. Journal of Open Source Software, 5(51), p.2272.

D’Agostino McGowan L., and Bryan J., 2022. googledrive: An Interface to Google Drive. https://googledrive.tidyverse.c https://github.com/tidyverse/googledrive.

Hijmans, R. J., Van Etten, J., Cheng, J., Mattiuzzi, M., Sumner, M., Greenberg, J. A., Lamigueiro, O. P., Bevan, A., Racine, E. B. & Shortridge, A. 2015. Package ‘raster’. R package, 734.

`extract_coords_combine` *Combine extracted explanatory variable data for occurrence records into single data frame.*

Description

Combines the split output files from functions extract_dynamic_coords and extract_buffered_coords into single data frame containing all occurrence records and explanatory variables.

Usage

```
extract_coords_combine(varnames, local.directory)
```

Arguments

`varnames` a character string, the unique names for each explanatory variable.

`local.directory` a character string or vector, the path to local directory or directories to read extracted explanatory data frames from.

Details

When extract_dynamic_coords and extract_buffered_coords have been used to extract dynamic explanatory variables for occurrence records, output for individual records and each variable will be split into separate ".csv" files.

This function reads in these files and combined data into a single data frame containing each occurrence records and associated explanatory data from each variable.

To prevent error, the ".csv" files must be uniquely named within the folder and include an exact character match for the varnames provided. All ".csv" files matching the varnames should have the same number and names of columns, which the default output of extract_dynamic_coords and extract_buffered_coords will if given the same occ.data data frame.

Value

Returns a data frame containing all occurrence records with associated explanatory variable data.

---

extract_dynamic_coords *Extract temporally dynamic explanatory variable data for occurrence records.*

Description

For each species occurrence record co-ordinate and date, temporally dynamic explanatory data are extracted using Google Earth engine

Usage

```
extract_dynamic_coords(
  occ.data,
  datasetname,
  bandname,
  spatial.res.metres,
  GEE.math.fun,
  save.method,
  temporal.res,
  temporal.direction,
  varname,
  save.directory
)
```

Arguments

`occ.data`	a data frame, with columns for occurrence record co-ordinates and dates with column names as follows; record longitude as "x", latitude as "y", year as "year", month as "month", and day as "day".
`datasetname`	a character string, the Google Earth Engine dataset to extract data from.
`bandname`	a character string, the Google Earth Engine dataset bandname to extract data for.
`spatial.res.metres`	a numeric value, the spatial resolution in metres for data extraction.
`GEE.math.fun`	a character string, the mathematical function to compute across the temporal.res period for each record.
`save.method`	a character string, the method used to save extracted variable data. One of '"split"' or '"combined"': can be abbreviated. See details.
`temporal.res`	a numeric value, the temporal resolution in days to extract data and calculate GEE.math.fun across from record date.
`temporal.direction`	a character string, the temporal direction for extracting data across relative to the record date. One of '"prior"' or '"post"': can be abbreviated.
`varname`	optional; a character string, the unique name for the explanatory variable. Default varname is "bandname_temporal.res_temporal.direction_ GEE.math.fun".
`save.directory`	a character string, the path to a local directory to save extracted variable data to.

Details

'For each individual species occurrence record co-ordinate and date, this function extracts data for a given band within a Google Earth Engine dataset across a user-specified period and calculates a mathematical function on such data.

GEE.math.fun specifies the mathematical function to be calculated over the temporal period. Options are limited to Google Earth Engine ImageCollection Reducer functions (https://developers.google.com/earth-engine/apidocs/) for which an analogous R function is available. This includes: "allNonZero","anyNonZero", "count", "first","firstNonNull", "last", "lastNonNull", "max","mean", "median","min", "mode","product", "sampleStdDev", "sampleVariance", "stdDev", "sum" and "variance".

For save.method '"combined"', the function with save ".csv" files containing all occurrence records and associated values for the explanatory variable. If save.method '"split"' is chosen, the function will save individual ".csv" files for each record with assigned unique ID value in file name. '"split"' method is provided to protect users if internet connection is lost when extracting data for large occurrence datasets.

Value

Returns details of successful explanatory variable extractions.

References

Aybar, C., Wu, Q., Bautista, L., Yali, R. and Barja, A., 2020. rgee: An R package for interacting with Google Earth Engine. Journal of Open Source Software, 5(51), p.2272.

`extract_dynamic_raster` *Extract temporally dynamic rasters of explanatory variable data.*

Description

Extract rasters for temporally dynamic explanatory variables at each projection date using Google Earth Engine.

Usage

```
extract_dynamic_raster(
  dates,
  spatial.ext,
  datasetname,
  bandname,
  spatial.res.metres,
  GEE.math.fun,
  user.email,
  varname = NULL,
  temporal.res,
  temporal.direction,
  save.directory = NULL,
  save.drive.folder
)
```

Arguments

`dates`	a character string, vector of dates in format YYYY-MM-DD.
`spatial.ext`	the spatial extent for the extracted raster. Object of class "Extent", "RasterLayer" or "polygon" or numeric vector listing xmin, xmax, ymin and ymax in order.
`datasetname`	a character string, the Google Earth Engine dataset to extract data from.
`bandname`	a character string, the Google Earth Engine dataset bandname to extract data for.
`spatial.res.metres`	a numeric value, specifying the spatial resolution in metres of the raster to be extracted.
`GEE.math.fun`	a character string, the mathematical function to compute across the specified time frame from each projection date and for each cell.
`user.email`	a character string, user email for initialising Google Drive.
`varname`	optional; a character string, the unique name for the explanatory variable. Default varname is “bandname_temporal.res_temporal.direction_ GEE.math.fun_raster".
`temporal.res`	a numeric value, the temporal resolution in days to extract data across.
`temporal.direction`	a character string, the temporal direction for extracting dynamic variable data across relative to each projection date given. One of ’"prior"’ or ’"post"’: can be abbreviated.
`save.directory`	optional; a character string, path to local directory to save extracted rasters to.
`save.drive.folder`	a character string, Google Drive folder to save extracted rasters to.

Details

For each projection date, this function downloads rasters at given spatial extent and resolution for temporally dynamic explanatory variables. For each cell in spatial extent, the GEE.math.fun is calculated for data extracted across specified number of days prior or post the projection date. Rasters of such data are saved directly to Google Drive, with option to export to local directory too. These can be combined to create projection covariate data frames for projection dynamic species distribution and abundance at high spatiotemporal resolution

extract_dynamic_raster requires users to have installed R package "rgee" (Aybar et al., 2020) and initialised Google Earth Engine with valid log-in credentials. Please follow instructions on the following website . datasetname must be in the accepted Google Earth Engine Data catalogue layout (e.g. “MODIS/006/MCD12Q1” or “UCSB-CHG/CHIRPS/DAILY”) and bandname as specified in the dataset (e.g. “LC_Type5”, “precipitation”). For datasets and band names, see .

extract_dynamic_raster also requires users to have installed the R package "googledrive" (D’Agostino McGowan and Bryan, 2022) and initialised Google Drive with valid log-in credentials, which must be provided under argument user.email. Please follow instructions on https://googledrive.tidyverse.org/ for initialising the googledrive package.

GEE.math.fun specifies the mathematical function to be calculated over the temporal period from each projection date. Options are limited to Google Earth Engine ImageCollection Reducer functions (https://developers.google.com/earth-engine/apidocs/) for which an analogous R function is available. This includes: "allNonZero","anyNonZero", "count", "first","firstNonNull", "last", "lastNonNull", "max","mean", "median","min", "mode","product", "sampleStdDev", "sampleVariance", "stdDev", "sum" and "variance".

Value

Returns details of successful explanatory variable extractions for each projection date.

References

Aybar, C., Wu, Q., Bautista, L., Yali, R. and Barja, A., 2020. rgee: An R package for interacting with Google Earth Engine. Journal of Open Source Software, 5(51), p.2272.

D’Agostino McGowan L., and Bryan J., 2022. googledrive: An Interface to Google Drive. https://googledrive.tidyverse.o https://github.com/tidyverse/googledrive.

`extract_xy_min_max` *extract_xy_min_max Extracts xmin,xmax,ymin,ymax from spatial extent object*

Description

extract_xy_min_max Extracts xmin,xmax,ymin,ymax from spatial extent object

Usage

```
extract_xy_min_max(x)
```

Arguments

x	spatial extent object. Object of class "Extent", "raster" or "polygon" or numeric vector listing xmin, xmax, ymin and ymax in order.

get_moving_window *Generate a “moving window” matrix of optimal size*

Description

Calculates an optimal “moving window” matrix size for use when extracting spatially buffered explanatory variables, by using the radius of interest and spatial resolution of environmental data.

Usage

```
get_moving_window(
  radial.distance,
  spatial.res.degrees = NULL,
  spatial.res.metres = NULL,
  spatial.ext = NULL
)
```

Arguments

`radial.distance`
: a numeric value, the radius of interest in metres.

`spatial.res.degrees`
: a numeric value, the spatial resolution in degrees of explanatory variable data.

`spatial.res.metres`
: a numeric value, the spatial resolution in metres of explanatory variable data.

`spatial.ext`
: an object of class "Extent", "raster" or "polygon" or numeric vector listing xmin, xmax, ymin and ymax in order, the spatial extent of study.

Details

To extract spatially buffered explanatory variable data using dynamicSDM functions extract_buffered_coords or extract_buffered_raster, a “moving window” matrix specifying the neighbourhood of cells to include in the calculation is required. For example, by using a three by three “moving window” matrix of equal weights, the explanatory variable would be calculated across the nine grid cells neighbouring the cell of interest and the cell of interest. The benefit of using a “moving window” over calculating explanatory variable values across a set radius from each record co-ordinate, is that when generating projection rasters at high spatial and temporal resolution, these can be generated much faster as the “moving windows” standardise the calculation. To calculate the “moving window” matrix size, the get_moving_window function first calculates the circular area of interest, using the user-specified radius of interest and equation for area of a circle. This radius should be chosen to represent the radial distance from species occurrence record co-ordinates that the explanatory variable data might be relevant and impact species presence. Then, the average grid cell area of the explanatory variable data (derived from user-provided spatial resolution and extent) is calculated. If spatial.res.degrees is given then spatial.ext is required to calculate average cell area size. If spatial.res.metres is given then average cell area is calculated by squaring this value to get cell area in square metres.

The function then calculates the optimal “moving window” matrix that best matches circular area of interest with the “moving window” matrix area.

Value

Returns "moving window" matrix of equal weights.

Examples

```
get_moving_window(radial.distance=100000,spatial.res.metres=111320)
```

---

sample_model_data — *Sample species occurrence records with associated dynamic explanatory variables*

Description

A dataset containing a sample of the bird species, the red-billed quelea (Quelea quelea), distribution and abundance records from between 2002-2017 including generated pseudo-absence records and associated extracted dynamic explanatory variables. The variables are as follows:

Usage

```
sample_model_data
```

Format

A data frame with 10000 rows and 14 variables:

x species occurrence record longitude.

y species occurrence record latitude.

year species occurrence record year.

month species occurrence record month.

day species occurrence record day.

presence.absence binary species presence or absence at record location and date.

individualCount number of individuals of the species present at record location and date.

sampling_weights total number of avian e-Bird sampling events within spatiotemporal buffer of occurrence record location and dates.

Temperaturemean mean MODIS Terra Land Surface Temperature at record co-ordinate across 52-weeks prior to record date (kelvin).

TemperatureEightmean mean MODIS Terra Land Surface Temperature at record co-ordinate across 8-weeks prior to record date (kelvin).

Precipitationsum sum Climate Hazards Group InfraRed Precipitation With Station Data (CHIRPS Daily) total daily precipitation at record co-ordinate across 52-weeks prior to record date (mm).

Precipitation8Wsum sum Climate Hazards Group InfraRed Precipitation With Station Data (CHIRPS Daily) total daily precipitation at record co-ordinate across 52-weeks prior to record date (mm).

watercount total number of MODIS Land Cover Type Yearly 500m " Water Bodies: at least 60% of area is covered by permanent water bodies" cells in surrounding area of record co-ordinate in record year.

blockno Block group assinged for model fitting by jacknife approach to account for spatial and temporal autocorrelation

References

Friedl, M., Sulla-Menashe, D. (2019). MCD12Q1 MODIS/Terra+Aqua Land Cover Type Yearly L3 Global 500m SIN Grid V006 Data set. NASA EOSDIS Land Processes DAAC. Accessed 2022-11-24 from https://doi.org/10.5067/MODIS/MCD12Q1.006

Funk, Chris, Pete Peterson, Martin Landsfeld, Diego Pedreros, James Verdin, Shraddhanand Shukla, Gregory Husak, James Rowland, Laura Harrison, Andrew Hoell & Joel Michaelsen. "The climate hazards infrared precipitation with stations-a new environmental record for monitoring extremes". Scientific Data 2, 150066. doi:10.1038/sdata.2015.66 2015.

GBIF.org (12 July 2021) GBIF Occurrence Download https://doi.org/10.15468/dl.ppcu6q

GBIF.org (25 July 2022) GBIF Occurrence Download https://doi.org/10.15468/dl.k2kftv

Wan, Z., Hook, S., Hulley, G. (2015). MOD11A1 MODIS/Terra Land Surface Temperature/Emissivity Daily L3 Global 1km SIN Grid V006 Data set. NASA EOSDIS Land Processes DAAC. Accessed 2022-11-24 from https://doi.org/10.5067/MODIS/MOD11A1.006

`sample_occ_abs_data` *Sample species occurrence records with pseudo-absence records*

Description

A dataset containing a sample of the bird species, the red-billed quelea (Quelea quelea), distribution and abundance records between 2002-2017 with generated spatiotemporal pseudo-absence records. The variables are as follows:

Usage

```
sample_occ_abs_data
```

Format

A data frame with 50 rows and 7 variables:

x species occurrence record longitude.

y species occurrence record latitude.

year species occurrence record year.

month species occurrence record month.

day species occurrence record day.

presence.absence binary species presence or absence at record location and date.

individualCount number of individuals of the species present at record location and date.

References

GBIF.org (12 July 2021) GBIF Occurrence Download https://doi.org/10.15468/dl.ppcu6q

GBIF.org (25 July 2022) GBIF Occurrence Download https://doi.org/10.15468/dl.k2kftv

`sample_occ_data` *Sample species occurrence records*

Description

A dataset containing a sample of the bird species, the red-billed quelea (Quelea quelea), distribution and abundance records between 2002-2017. The variables are as follows:

Usage

```
sample_occ_data
```

Format

A data frame with 10000 rows and 14 variables:

x species occurrence record longitude.

y species occurrence record latitude.

year species occurrence record year.

month species occurrence record month.

day species occurrence record day.

presence.absence binary species presence or absence at record location and date.

individualCount number of individuals of the species present at record location and date.

References

GBIF.org (12 July 2021) GBIF Occurrence Download https://doi.org/10.15468/dl.ppcu6q

GBIF.org (25 July 2022) GBIF Occurrence Download https://doi.org/10.15468/dl.k2kftv

`sample_surveyeffort` *Sample e-Bird sampling event records*

Description

A dataset containing a sample of e-Bird sampling events for all bird species across southern Africa between 2000-2020. The variables are as follows:

Usage

```
sample_surveyeffort
```

Format

A data frame with 10000 rows and 5 variables:

day avian e-Bird sampling event day.

month avian e-Bird sampling event month.

year avian e-Bird sampling event year.

y avian e-Bird sampling event latitude.

x avian e-Bird sampling event longitude.

References

Fink, D., T. Auer, A. Johnston, M. Strimas-Mackey, O. Robinson, S. Ligocki, W. Hochachka, L. Jaromczyk, C. Wood, I. Davies, M. Iliff, L. Seitz. 2021. eBird Status and Trends, Data Version: 2020; Released: 2021. Cornell Lab of Ornithology, Ithaca, New York. https://doi.org/10.2173/ebirdst.2020

GBIF.org (12 July 2021) GBIF Occurrence Download https://doi.org/10.15468/dl.ppcu6q

---

`skip_if_no_GEE_credentials` *skip_if_no_GEE_credentials Skips test_that test if no Google Earth Engine credentials set-up*

Description

skip_if_no_GEE_credentials Skips test_that test if no Google Earth Engine credentials set-up

Usage

```
skip_if_no_GEE_credentials()
```

`spatiotemp_autocorr` *Test for spatial and temporal autocorrelation in species distribution model explanatory data.*

Description

Function performs statistical tests to assess spatial and temporal autocorrelation in given explanatory variable data.

Usage

```
spatiotemp_autocorr(occ.data, varname, temporal.level)
```

Arguments

`occ.data`	a data frame, with columns for occurrence record co-ordinates and dates with column names as follows; record longitude as "x", latitude as "y", year as "year", month as "month", and day as "day" and associated explanatory data.
`varname`	a character string or vector, the name(s) of the columns within occ.data containing data to test for autocorrelation.
`temporal.level`	a character string, the time step to test for temporal autocorrelation at. One of '"day"' or '"month"', '"year"'. Can be abbreviated.

Details

To test for temporal autocorrelation, the function first calculates the average value across records for each time step (temporal.level). The correlation between the average value at one time point (t) and the value at the previous time point (t-1) is calculated and plotted. A significant relationship between values at consecutive data points indicates temporal autocorrelation is present.

To test for spatial autocorrelation, the function calculates a distance matrix between all record co-ordinates. Moran's I statistical test is calculated to test whether points closer in space have more similar values than those more distant from each other.

As the spatial autocorrelation calculation involves computation of a distance matrix between all occurrence records. To reduce computation time, it is recommended that a sample of large occurrence datasets are input.

Value

Returns a list of temporal and spatial autocorrelation test results for each variable.

Examples

```
data("sample_model_data")
spatiotemp_autocorr(sample_model_data,varname="Temperaturemean",temporal.level="year")
```

---

spatiotemp_bias — *Test for spatial and temporal bias in species occurrence records*

Description

Generates plots for visual assessment of spatial and temporal biases in occurrence records. Tests whether the spatiotemporal distribution of records is significantly different from the distribution from random sampling.

Usage

```
spatiotemp_bias(occ.data, temporal.level)
```

Arguments

`occ.data`	a data frame, with columns for occurrence record co-ordinates and dates with column names as follows; record longitude as "x", latitude as "y", year as "year", month as "month", and day as "day".
`temporal.level`	a character string, the time step to test for temporal bias at. One of '"day"' or '"month"', '"year"'. Can be abbreviated.

Details

To assess temporal sampling bias, the function returns a histogram plot of the frequency distribution of records across the given time step specified by temporal.level. The observed frequency of sampling across the categorical time steps are compared to the distribution expected from random sampling, using a chi-squared test.

To assess spatial sampling bias, the function returns a scatterplot of the spatial distribution of occurrence records to illustrate any spatial clustering. The average nearest neighbour distance of record co-ordinates is then compared to that of records randomly generated at same density using a t-test.

As the spatial bias calculation involves calculation of a distance matrix. To reduce computation time, it is recommended that a sample of large occurrence datasets are input.

Value

Returns list containing chi-squared and t-test results.

Examples

```
data("sample_occ_data",package="dynamicSDM")
spatiotemp_bias(occ.data = sample_occ_data,temporal.level = "month")
```

---

spatiotemp_block   *Split occurrence records into spatial and temporal blocks for model fitting.*

Description

Splits occurrence records into spatial and temporal sampling units and groups sampling units into multiple blocks that have similar mean and range of environmental explanatory variables.

Usage

```
spatiotemp_block(
  occ.data,
  vars.to.block.by,
  spatial.layer = NULL,
  spatial.split.degrees = NULL,
  temporal.block = NULL,
  n.blocks = 10,
  iterations = 5000
)
```

Arguments

`occ.data`	a data frame, with columns for occurrence record co-ordinates and dates with column names as follows; record longitude as "x", latitude as "y", year as "year", month as "month", and day as "day", and associated explanatory variable data.
`vars.to.block.by`	a character string or vector, the explanatory variable column names to group sampling units based upon.
`spatial.layer`	optional; a RasterLayer object, a categorical spatial layer for sampling unit splitting.
`spatial.split.degrees`	optional; a numeric value, the grid cell resolution in degrees to split spatial.layer by.
`temporal.block`	optional; a character string or vector, the time step for sampling unit splitting. Any combination of ’"day"’, ’"month"’ or ’"year"’ or "quarter".
`n.blocks`	optional; a numeric value, the number of blocks to group occurrence records into. Default; 10.
`iterations`	optional; a numeric value, the number of random block groupings to trial before selecting the optimal grouping. Default; 5000.

Details

Blocking is an established method to account for spatial autocorrelation in SDMs. Following Bagchi et al., (2013), the blocking method involves splitting occurrence data into sampling units based upon non-contiguous ecoregions, which are then grouped into spatially disaggregated blocks of approximately equal sample size, within which the mean and range of explanatory variable data are similar. When species distribution model fitting, blocks are left out in-turn in a jack-knife approach for model training and testing.

We adapt this approach to account for temporal autocorrelation by enabling users to split records into sampling units based upon spatial and temporal characteristic before blocking occurs.

Spatial splitting: If the spatial.layer raster has categories that take up large contiguous areas, spatial.split.degrees will split categories into smaller units using grid cells at given resolution. Temporal splitting: If temporal.block is given, then occurrence records with unique values for given level are considered unique sampling unit. For instance, if temporal.block = “year”, then records from the same year are considered a sampling unit to be grouped into blocks. However, if spatial splitting is also used, then spatial characteristics may split these further into separate sampling units. Once split into sampling units based upon temporal and spatial characteristics, these units are then assigned into given number of blocks (n.blocks), so that the mean and range of explanatory variables (vars.to.block.by) are similar across each.

Value

Returns occurrence data frame with column "BLOCK.CATS", assigning each record to a spatiotemporal block.

References

Bagchi, R., Crosby, M., Huntley, B., Hole, D. G., Butchart, S. H. M., Collingham, Y., Kalra, M., Rajkumar, J., Rahmani, A. & Pandey, M. 2013. Evaluating the effectiveness of conservation site networks under climate change: accounting for uncertainty. Global Change Biology, 19, 1236-1248.

Examples

```
data("sample_model_data")
data("biome_layer")
spatiotemp_block(
 occ.data = sample_model_data,
 spatial.layer = biome_layer,
 spatial.split.degrees = 3,
 temporal.block = c("month"),
 vars.to.block.by = colnames(sample_model_data)[9:12],
 n.blocks = 10
)
```

`spatiotemp_check` *Check species occurrence record formatting, completeness and validity.*

Description

Checks the occurrence record data frame contains the column names and classes required for dynamicSDM functions. Option to exclude records containing missing, duplicate or invalid co-ordinates or dates.

Usage

```
spatiotemp_check(
  occ.data,
  na.handle = "exclude",
  duplicate.handle = "exclude",
  coord.handle = "exclude",
  date.handle = "exclude"
)
```

Arguments

`occ.data`	a data frame, with columns for occurrence record co-ordinates and dates with column names as follows; record longitude as "x", latitude as "y", year as "year", month as "month", and day as "day".
`na.handle`	a character string, method for handling missing data (NA values) in record co-ordinates and dates. One of '"exclude"' or '"ignore"': can be abbreviated. Default; exclude.
`duplicate.handle`	a character string, method for handling duplicate record co-ordinates or dates. One of '"exclude"' or '"ignore"': can be abbreviated. Default; exclude.
`coord.handle`	a character string, method for handling invalid co-ordinates in record data. One of '"exclude"' or '"ignore"': can be abbreviated. Default; exclude.
`date.handle`	a character string, method for handling invalid dates in record data. One of '"exclude"' or '"ignore"': can be abbreviated. Default; exclude.

Details

Record dates and co-ordinates are checked for validity using the following rules. Dates must be real dates that could exist. For example, 50th February 2000 is not a valid date. Co-ordinates must have longitude (x) values between -180 and 180, and latitude (y) values between -90 and 90 to be considered valid.

Value

By default, returns occurrence record data frame, filtered to exclude records containing missing, duplicate or invalid data in record co-ordinates and dates.

Examples

```
data("sample_occ_data")
spatiotemp_check(sample_occ_data)
```

spatiotemp_extent *Filter species occurrence records by a given spatial and temporal extent.*

Description

Function excludes species occurrence records with co-ordinates outside given spatial extent and dates outside given temporal extent.

Usage

```
spatiotemp_extent(occ.data, temporal.ext = NULL, spatial.ext = NULL)
```

Arguments

occ.data	a data frame, with columns for occurrence record co-ordinates and dates with column names as follows; record longitude as "x", latitude as "y", year as "year", month as "month", and day as "day".
temporal.ext	optional; a character vector, two dates in format YYYY-MM-DD. First date represents start of temporal extent and second date represents end of temporal extent for inclusion.
spatial.ext	optional; object of class "Extent", "raster" or "polygon" or numeric vector listing xmin, xmax, ymin and ymax in order, the spatial extent to filter by.

Details

If spatial.ext is provided, the function checks whether species occurrence record co-ordinates are within the given spatial extent of the study (spatial.ext) and excludes any outside of this extent.

If temporal.ext is provided, the function checks whether species occurrence record dates are within the given temporal extent of the study (temporal.ext) and excludes any outside of this extent.

Value

Returns data frame of occurrence records filtered to the spatial and temporal extent given.

Examples

```
x<-c(27.79125, 28.54125, 25.54125, 30.04125, 29.95792)
y<-c(-26.79125, -26.37458, -26.70792, -29.37458, -28.45792)
month<-c(1, 2, 3, 2, 4)
day<-c(27, 25, 16, 25, 26)
year<-c(2014, 2016, 2011, 2011, 2015)
occ.data<-data.frame(cbind(x,y,year,month,day))
spatiotemp_extent(occ.data,
                 temporal.ext=c("2012-01-01","2017-01-01"),
                spatial.ext =c(28,31,-30,-26))
```

spatiotemp_pseudoabs *Generate pseudo-absence record coordinates and dates*

Description

Function generates specified number of pseudo-absence record co-ordinates and dates either randomly or buffered in space and time.

Usage

```
spatiotemp_pseudoabs(
  spatial.method,
  temporal.method,
  occ.data = NULL,
  spatial.ext = NULL,
  temporal.ext = NULL,
  spatial.buffer = NULL,
  temporal.buffer = NULL,
  n.pseudoabs = 100
)
```

Arguments

`spatial.method`	a character string, the spatial method for pseudo-absence generation. One of '"buffer"' or '"random"': can be abbreviated.
`temporal.method`	a character string, the temporal method for pseudo-absence generation. One of '"buffer"' or '"random"': can be abbreviated.
`occ.data`	optional; a data frame, with columns for occurrence record co-ordinates and dates with column names as follows; record longitude as "x", latitude as "y", year as "year", month as "month", and day as "day". Required if either temporal.method or spatial.method is "'buffer'".
`spatial.ext`	optional; object of class "Extent", "raster" or "polygon" or numeric vector listing xmin, xmax, ymin and ymax in order, the spatial extent to randomly generate pseudo-absences co-ordinates within.
`temporal.ext`	optional; a character vector, two dates in format YYYY-MM-DD. First represents start of temporal extent and second represents end of temporal extent to randomly generate pseudo-absences dates within. Only required if temporal.method is "'random'".
`spatial.buffer`	optional; a numeric vector, the distances in metres of radii to generate buffered pseudo-absence coordinates within. Only required if spatial.method is "'buffer'".
`temporal.buffer`	optional; a numeric vector, the two periods in days to generate buffered pseudo-absence dates within. Only required if temporal.method is "'buffer'".
`n.pseudoabs`	optional; a numeric value, the number of pseudo-absence records to generate. Default; 100.

Details

If spatial.method is "'buffer'", then occurrence record co-ordinates are randomly generated between the given closest radius and furthest away radius specified in spatial.buffer. For example, if “spatial. buffer = c(3000,10000)”, then pseudo-absence co-ordinates are randomly generated at least 3000m radius away from occurrence record co-ordinate but within 10000m radius.

If spatial.method is "'random'", then occurrence record co-ordinates are randomly generated across spatial.ext object given.

If temporal.method is "'buffer'", then occurrence record dates are randomly generated between the closest and further away number of days specified in temporal.buffer. For example, if “temporal buffer = c(14,30)”, then pseudo-absence dates randomly generated at least 14 days from occurrence record dates but within 30 days.

If temporal.method is "'random'", then occurrence record co-ordinates are randomly generated within the two temporal.ext dates given.

Value

Returns data frame of pseudo-absence coordinates and dates.

Examples

```
data("sample_occ_data")
spatiotemp_pseudoabs(
 sample_occ_data,
 spatial.method = "random",
 temporal.method = "random",
 spatial.ext = c(20, 36, -35, -12),
 temporal.ext = c("2011-01-01", "2017-01-01")
)
```

`spatiotemp_resolution` *Filter species occurrence records by given spatial and temporal resolution*

Description

Filters species occurrence record data frame to exclude records with co-ordinates and dates that do not meet specified spatial and temporal resolution.

Usage

```
spatiotemp_resolution(occ.data, spatial.res = NULL, temporal.res = NULL)
```

Arguments

`occ.data`	a data frame, with columns for occurrence record co-ordinates and dates with column names as follows; record longitude as "x", latitude as "y", year as "year", month as "month", and day as "day".
`spatial.res`	optional; a numeric value, the minimum acceptable number of decimal places given for occurrence record co-ordinates.
`temporal.res`	optional; a character string, the minimum acceptable temporal resolution of occurrence record dates. One of '"day"', '"month"' or '"year"': can be abbreviated.

Details

Excludes species occurrence records that do not meet the minimum spatial and temporal resolution specified.

If spatial.res given, the value of 1 represents acceptable co-ordinate resolution of one decimal place, roughly equal to 11.1km, and value of 3 represents three decimal places, roughly equal to 111m.

If temporal.res given, temporal.res = "day" would result in exclusion of records without values for year, month and day, and temporal.res = "year" would only exclude records without values for year.

spatial.res and temporal.res can be informed based upon the highest spatial and temporal resolution of the datasets to be utilised when extracting dynamic variables. For instance, if explanatory variables datasets are annual, then temporal.res of "year" is adequate, whereas if datasets are daily, then temporal.res of "day" is more appropriate.

Value

Returns a data frame of species records filtered by the minimum acceptable spatial resolution of co-ordinates and temporal resolution of dates.

Examples

```
x<-c(27.1, 28.54125, 25.54125, 30.04125, 29.95792)
y<-c(-26.79125, -26.37458, -26.70792, -29.37458, -28.45792)
year<-c(2014, 2016, 2011, 2011, 2015)
month<-c(1, 2, 3, 2, 4)
day<-c(27, 25, 16, 25, 26)
occ.data<-data.frame(cbind(x,y,year,month,day))
spatiotemp_resolution(occ.data,spatial.res=5,temporal.res="day")
```

`spatiotemp_thin` *Thin species occurrence records by spatial and temporal proximity.*

Description

Thins species occurrence records that are within minimum spatial and temporal distance apart.

Usage

```
spatiotemp_thin(
  occ.data,
  temporal.method,
  temporal.dist,
  spatial.split.degrees,
  spatial.dist,
  iterations = 100
)
```

Arguments

`occ.data`	a data frame, with columns for occurrence record co-ordinates and dates with column names as follows; record longitude as "x", latitude as "y", year as "year", month as "month", and day as "day".

`temporal.method`
: a character string, the method to calculate temporal distance between records. One of "'DOY'" or "'day'". See details for more information.

`temporal.dist`
: a numeric value, the temporal buffer in days to thin records by.

`spatial.split.degrees`
: a numeric value, the grid cell resolution in degrees to split occurrence records by before temporal thinning.

`spatial.dist`
: a numeric value, the spatial buffer distances in metres to thin records by.

`iterations`
: a numeric value, the number of iterations to randomly thin occurrence records by. Default; 100.

Details

For temporal thinning, the function first splits occurrence records into grid cells of given size in degrees (set by spatial.split.degrees). This is to prevent spatially distant but temporally close records from being excluded. For each grid cell, all records within the cell are temporally thinned. This process works by removing records that are within given temporal distance (temporal.dist) from each other by randomly selecting one of the two. This iterates through until no records are within the given temporal distance of each other in each grid cell, following a similar algorithm to spThin (Aiello-Lammens et al., 2015).

Two methods exist for measuring the temporal distance between occurrence records. temporal.method = “DOY” calculates the minimum days apart within the annual cycle and “day” uses the absolute number of days. For instance, two dates “2010-01-05” and “2012-12-05” can be calculated as either 1065 absolute days apart, or within the annual cycle these dates represent day 5 and day 339 of the year, and are 31 days apart. Therefore, thinning by 40 days using “DOY” method would remove one of these records, but using the “day” method would not. Chosen temporal.method will depend upon whether bias towards a point within the annual cycle is more likely to bias models than bias towards a given point in linear time.

Following temporal thinning, spatial thinning occurs across entire dataset. The spatial distance between each record is calculated, and records within the given spatial distance (spatial.dist) from each other are excluded by randomly selecting one of these. This iterates through until no records are with the given spatial distances of each other across entire dataset, again following a similar algorithm to spThin (Aiello-Lammens et al., 2015).

As random selection could alter the total number of occurrence records remaining in the occurrence record dataset, this process is iterated through a specified number of times and the thinned data frame with the highest number of records remaining is returned.

Value

Returns data frame of occurrence records thinned by specified temporal and spatial distance.

References

Aiello-Lammens, M. E., Boria, R. A., Radosavljevic, A., Vilela, B. & Anderson, R. P. 2015. spThin: an R package for spatial thinning of species occurrence records for use in ecological niche models. Ecography, 38, 541-545.

Examples

```
x<-c(27.79125, 28.54125, 25.54125, 30.04125, 29.95792)
y<-c(-26.79125, -26.37458, -26.70792, -29.37458, -28.45792)
year<-c(2014, 2016, 2011, 2011, 2015)
```

```
month<-c(1, 2, 3, 2, 4)
day<-c(27, 25, 16, 25, 26)
occ.data<-data.frame(cbind(x,y,year,month,day))

spatiotemp_thin(
 occ.data = occ.data,
 temporal.method = "day",
 temporal.dist = 14,
 spatial.split.degrees = 3,
 spatial.dist = 100000,
 iterations = 5
)
```

`spatiotemp_weights` *Calculate sampling effort across spatial and temporal buffer from species occurrence records*

Description

Calculates the total number of sampling events across a given spatial and temporal buffer from each occurrence record's co-ordinate and date.

Usage

```
spatiotemp_weights(
  occ.data,
  sampling.events.df,
  spatial.dist = NULL,
  temporal.dist = NULL
)
```

Arguments

`occ.data`	a data frame, with columns for occurrence record co-ordinates and dates with column names as follows; record longitude as "x", latitude as "y", year as "year", month as "month", and day as "day".
`sampling.events.df`	a data.frame, sampling events with column names as follows; record longitude as "x", latitude as "y", year as "year", month as "month", and day as "day".
`spatial.dist`	a numeric value, the spatial distance in metres representing the radius from occurrence record co-ordinate to sum sampling events across.
`temporal.dist`	a numeric value, the temporal distance in days, representing the period before and after the occurrence record date to sum sampling events across.

Details

For each occurrence record, this function calculates the total number of sampling events within given radius (spatial.dist) from each record co-ordinate and days (temporal.dist) both prior and post record date.

In addition to total sampling events, the function also calculates relative sampling effort, scaling from 0 (least sampled) to 1 (most sampled).

Output could be used as model weights to correct spatial and temporal biases in occurrence record collections.

Value

Returns input occurrence record data frame with additional columns for sampling effort "SAMP_EFFORT" and relative sampling effort "REL_SAMP_EFFORT".

Examples

```
data("sample_occ_abs_data")
data("sample_surveyeffort")
spatiotemp_weights(
 occ.data = sample_occ_abs_data,
 sampling.events.df = sample_surveyeffort,
 spatial.dist = 200000,
 temporal.dist = 20
)
```

Index