Package: sdmtools (via r-universe)

October 30, 2024

Title Utility tools for Species Distribution Modelling

Version 0.0.0.9000

Description What the package does (one paragraph).

License MIT + file LICENSE

URL `https://github.com/idem-lab/sdmtools`

BugReports `https://github.com/idem-lab/sdmtools/issues`

Depends R (>= 4.1.0)

Imports dplyr, grDevices, gtools, magrittr, malariaAtlas, multispeciesPP (>= 1.0), purrr, sf, stats, terra, tibble, tidyr, tidyselect

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Suggests testthat (>= 3.0.0)

Remotes wfithian/multispeciesPP

Config/testthat/edition 3

Repository https://idem-lab.r-universe.dev

RemoteUrl https://github.com/idem-lab/sdmtools

RemoteRef HEAD

RemoteSha 03f6176d7819a9a8aadc3918eeae8dde7cd3a276

Contents

assign_nearest_land 2
example_mask 3
example_raster 3
example_vector 4
extract_covariates 4

global_regions 6
import_rasts 7
inside_mask 8
make_africa_mask 8
make_mpp_list 9
maskpointsdf 10
mask_all 11
mask_from_all 12
match_ref 13
nearest_land 13
new_long_tibble 14
predict_mpp_rast 14
predict_mpp_rast_all 15
predict_sdm 16
print.long_tibble 17
raster_to_terra 17
rastpointplot 18
save_plot 19
set_layer_names 20
set_levels 20
source_R 21
split_rast 22
standardise_rast 23
std_rast 23
temptif 24
writereadrast 24

Index **26**

`assign_nearest_land` *Assign to nearest raster cell on mask*

Description

Adapted from seegSDM. Reposition observations to a location within mask if within a specified distance. Useful for when coastal observations drop off jagged mask and similar

Usage

```
assign_nearest_land(dat_object, mask_object, max_distance, verbose = TRUE)
```

Arguments

`dat_object`	data.frame
`mask_object`	'SpatRaster'
`max_distance`	map units if raster is projected
`verbose`	provide verbose output. Default is TRUE

Value

modified data object minus observations not within specified maximum distance of mask

---

`example_mask` *Create an example mask within a given raster*

Description

Create an example mask within a given raster

Usage

```
example_mask(raster, pc_threshold = NULL)
```

Arguments

`raster`	generated from example_raster for example
`pc_threshold`	A threshold percentile to divide mask / non mask elements

Value

A `spatRaster`

Examples

```
example_mask(example_raster(), pc_threshold=0.5)
```

`example_raster` *Create example raster for use in examples*

Description

Create example raster for use in examples

Usage

```
example_raster(seed = NULL, layername = NULL)
```

Arguments

`seed`	A seed to pass to `set.seed(seed)`
`layername`	`character` for layer name passed to `names`

Value

A `SpatRaster`.

Examples

```
example_raster()

example_raster(
  seed = 3.142,
  layername = "jabberwock_density"
)
```

example_vector *Create an example vector*

Description

Create an example vector

Usage

```
example_vector(seed = NULL)
```

Arguments

seed	A seed to pass to `set.seed(seed)`

Value

A `SpatVector`

Examples

```
example_vector()
```

extract_covariates *Extract data from covariate rasters*

Description

Extracts data from raster covariate layers for modelling.

Usage

```
extract_covariates(
  covariates,
  presences = NULL,
  absences = NULL,
  presences_and_absences = NULL
)
```

Arguments

`covariates` `SpatRaster` object of one or more layers.

`presences` `data.frame` or `tibble` of presence locations contaning longitude and latitude as variables `x` and `y`.

`absences` `data.frame` or `tibble` of absence or background locations contaning longitude and latitude as variables `x` and `y`.

`presences_and_absences`
`data.frame` or `tibble` of presence presence and absence locations containing longitude and latitude as variables `x` and `y`.

Details

`extract_covariates` will run correctly with either `presences` only, `presences` and `absences`, or `presences_and_absences` only. (If all three are included it will ignore the last one).

Value

A `tibble` containing values variables `presence` and each of the covariate layers at each location.

Author(s)

Gerry Ryan

Examples

```
library(terra)

cov1 <- example_raster(
 seed = -44,
 layername = "cov1"
)
cov2 <- example_raster(
 seed = 15.3,
 layername = "cov2"
)

covs <- c(cov1, cov2)


presences <- example_vector(seed = 68) %>%
 as.data.frame(geom = "xy")
absences <- example_vector(seed = 9.6) %>%
 as.data.frame(geom = "xy")

extract_covariates(
 covariates = covs,
 presences = presences,
 absences = absences
)
```

`global_regions` *Global regions*

Description

A table of showing the United Nations Regional groups of Member States and World Health Organisation regions of nation states and all states with Officially Assigned ISO3166 Alpha-3 country codes.

Usage

```
global_regions
```

Format

`global_regions` **A tibble with 249 rows and 6 columns::**

country Country name. Given name conflicts between the two data sources, names identified in `country` are defined as `countrycode::countrycode(global_regions$iso3, origin = "iso3c", destination = "country.name")`

iso2, iso3 Alpha-2 & Alpha-3 ISO3166 country codes

who_region World Health Organisation region

un_region United Nations Regional group of Member States

continent Continent

Details

Includes all Officially Assigned ISO3166 Alpha-3 country codes.

Not all 'countries' include a continent.

Special cases of UN regional groupings

Israel

In May 2000, Israel became a full member of the Group of Western European and other States on a temporary basis (subject to renewal), thereby enabling it to put forward candidates for election to various bodies of the General Assembly. In 2004, Israel obtained a permanent renewal to its membership.

Kiribati

As of 2010, Kiribati (geographically in Oceania) is not a member of any regional group, despite other Oceania nations belonging to the Group of Asia-Pacific States.

Türkiye

Türkiye participates fully in both the Group of Western European and other States and the Group of Asia-Pacific States, but for electoral purposes is considered a member of the Group of Western European and other States only.

While Türkiye is listed in both groupings in the original dataset; here we remove the duplication and list it only where it is a voting member, i.e., Western European and other States

United States of America

The United States of America is not a member of any regional group, but attends meetings of the Group of Western European and other States as an observer and is considered to be a member of that group for electoral purposes.

Source

Table combined from `un-regions.csv`, and `who-regions.csv`, housed in package data-raw directory.

`un-regions.csv` manually created from tables on "Regional groups of Member States"

https://www.un.org/dgacm/en/content/regional-groups

2024-02-02

`who-regions.csv` was downloaded from

https://ourworldindata.org/grapher/who-regions

2024-02-02

World Health Organization – processed by Our World in Data & World Health Organization.

`continent` defined by `countrycode::codelist$continent`, (largely?) by World Bank.

`import_rasts` *Import rasters from directory*

Description

Imports all rasters of a given extension type from a specified directory using `terra::rast`. Based on `seegSDM::importRasters`.

Usage

```
import_rasts(path, ext = ".grd", as_list = FALSE)
```

Arguments

`path`	Directory path containing rasters
`ext`	Extension type
`as_list`	`logical` Should the `spatRaster` objects be returned as a `list` (`TRUE`) or concatenated as layers in a single `SpatRaster` object (`FALSE`; default)

Value

A `SpatRaster` if `as_list` = `FALSE`, or list of `SpatRaster` objects.

Examples

```
## Not run:
  rasters <- import_rasts("/data/grids/covariates")

## End(Not run)
```

inside_mask *title*

Description

Checks whether longitude and latitude coincide with non-missing pixels of a raster. The function takes two arguments: points, a dataframe containing columns named longitude' and 'latitude', and mask is a raster. Returns a dataframe of longitude and latitude only those rows with points falling on non-missing pixels. If all points fall on missing pixels, the function throws an error.

Usage

```
inside_mask(points, mask)
```

Arguments

`points`	dataframe containing columns named 'longitude' and 'latitude'
`mask`	a raster

Value

dataframe of longitude and latitude only those rows with points falling on non-missing pixels. If all points fall on missing pixels, the function throws an error.

make_africa_mask *Make Africa Mask*

Description

Makes a `SpatRaster` or `SpatVector` mask layer of Africa, based on shapefiles for African nations from the `malaraAtlas` package.

Usage

```
make_africa_mask(
  filename = NULL,
  type = c("raster", "vector"),
  res = c("high", "low"),
  countries = NULL
)
```

Arguments

`filename`	Character of file path and name if mask is to be written to disc.
`type`	Character `raster` or `vector`; to return mask as either `SpatRaster` or `SpatVector`.
`res`	Character `"high"` or `"low"`; corresponding to resolution of 0.008333333 or 0.04166667 decimal degrees
`countries`	Character of ISO3 country names. If `NULL` returns all countries in Africa.

Details

Raster layers creates with extent of `terra::ext(-18.0000019073486, 52.0416647593181, -34.9999987284343, 37.5416679382324)`

Value

`SpatRaster` or `SpatVector` in WGS 84 (EPSG:4326).

Examples

```
## Not run:
# Create an object in workspace
africa_mask_v <- make_africa_mask(type = "vector")

# Save to disk
make_africa_mask(filename = "africa_mask.tif", type = "raster")

# or do both at once
africa_mask_r <- make_africa_mask("africa_mask.tif")

## End(Not run)
```

`make_mpp_list` *Make presence-only list for* `multispeciesPP`

Description

Make presence-only list for `multispeciesPP`

Usage

```
make_mpp_list(x, id)
```

Arguments

`x`	A `data.frame` containing covariate and bias values for locations, and an identity (species) column
`id`	The name of the identity column in `tidyselect` form. `id` must be a column name in `x`.

Details

The package multispeciesPP requires a very annoying named list format for presence-only data. This function takes a table of covariate / bias values and an ID column (i.e. species), and returns a named list where each element is the values corresponding to that identity only. Ugh.

Value

A named list or data frames.

Examples

```
## Not run:
make_mpp_list(presence_only_data, species)

## End(Not run)
```

maskpointsdf *Title*

Description

Title

Usage

```
maskpointsdf(df, msk)
```

Arguments

df	data.frame
msk	mask

Value

tibble of two columsn, lon and lat

mask_all *Mask all*

Description

Masks all NA cells across all layers, such that returned layers have matching NA cells.

Usage

```
mask_all(rasts, filename = NULL, overwrite = FALSE)
```

Arguments

rasts	SpatRaster with nlyr(rasts) > 1 to mask
filename	characterto save output
overwrite	logical overwrite existing filename?

Details

Uses a ton of RAM and will break for larger rasters.

Value

SpatRaster

Examples

```
# Create some SpatRaster layers with non-matching NA cells
library(terra)
library(sdmtools)

r <- c(
  example_raster(seed = 1),
  example_raster(seed = 2),
  example_raster(seed = 3)
)
rvals <- terra::values(r)
nas <- c(1:10, 105:120, 215:240)
rvals[nas] <- NA
r[] <- rvals

# check if it pleases you to do so
# plot(r)
# mask out non-overlapping `NA` values in all layers
s <- mask_all(r)
s

# plot(s)
```

mask_from_all *Create mask from raster layers*

Description

Creates a mask where a cell in any layer of `r` that is `NA` will be returned as `NA`.

Usage

```
mask_from_all(r)
```

Arguments

r	`SpatRaster` with >1 layer.

Details

Similar in intention to `mask_all`, but (a) will work on larger rasters because it only holds the values of a single layer in memory at a time, and (b) returns a mask layer, rather than masking each layer in `r`. **Can be very slow**

Value

`SpatRaster` with values `NA` or 1.

Examples

```
r <- example_raster(seed = 10)
s <- example_raster(seed = 11)

r[10:20] <- NA

s[5:15] <- NA

q <- mask_from_all(c(r,s))

library("terra")
plot(c(r,s,q))
```

`match_ref` *match ref*

Description

Crop, resample, and mask `x` against `ref`, and optionally replace any missing values.

Usage

```
match_ref(x, ref, missing_val = NULL, filename = NULL, overwrite = TRUE)
```

Arguments

`x`	`SpatRaster` object to bash into shape
`ref`	`SpatRaster` reference object
`missing_val`	If `NULL` missing value are left as is. Otherwise any `NA` or `NaN` values in `x` that are not `NA` in ref will be replaced by `missing_val`
`filename`	If not `NULL` output will be written to filename
`overwrite`	`logical`. If `filename` not `NULL`, then if `TRUE`, `filename` will be overwritten.

Value

`SpatRaster` of `nlyrs(x)` trimmed to extent and resolution of `ref`.

Examples

```
#placeholder example
```

`nearest_land` *Identify nearest land*

Description

Adapted from seegSDM Identify closest neighbouring cell that does not return NA on raster mask

Usage

```
nearest_land(points, raster, max_distance)
```

Arguments

`points`	anything `terra::extract()` accepts as the y argument
`raster`	raster
`max_distance`	the map units if raster is projected

Value

matrix of XY coordinates of nearest cell on mask, or NAs

new_long_tibble — long_tibble *constructor function*

Description

long_tibble constructor function

Usage

```
new_long_tibble(x)
```

Arguments

x	A tbl_df object.

Value

long_tibble.

predict_mpp_rast — *Raster predictions from* multispeciesPP

Description

Make spatial predictions from multispeciesPP::predict.multispeciesPP by passing in data as terra::SpatRaster layers and returning a SpatRaster.

Usage

```
predict_mpp_rast(
  model,
  data,
  sp,
  type = c("response", "link"),
  filename = NULL,
  overwrite = FALSE
)
```

Arguments

`model`	A `multispeciesPP` model object
`data`	`SpatRaster` object with covariate and bias layers
`sp`	`character` Species name to predict
`type`	`character`. Prediction scale — `"response"` or `"link"`.
`filename`	`character`to save output
`overwrite`	`logical` overwrite existing `filename`?

Value

`SpatRaster`

`predict_mpp_rast_all` *Raster predictions from* `multispeciesPP` *for all species*

Description

Calls `sdmtools::predict_mpp_rast` over all species in `model`

Usage

```
predict_mpp_rast_all(
  model,
  data,
  type = c("response", "link"),
  filename = NULL,
  overwrite = FALSE
)
```

Arguments

`model`	A `multispeciesPP` model object
`data`	`SpatRaster` object with covariate and bias layers
`type`	`character`. Prediction scale — `"response"` or `"link"`.
`filename`	`character`to save output
`overwrite`	`logical` overwrite existing `filename`?

Value

`SpatRaster`

predict_sdm *Predict Species Distribution Model*

Description

Produces raster prediction from SDM based on model and covariate layers.

Usage

```
predict_sdm(
  model,
  covariates,
  type = NULL,
  layer_name = "predicted_distribution"
)
```

Arguments

model	A model object.
covariates	SpatRaster covariate layers.
type	Scale of prediction (response, model, etc.).
layer_name	Name for predicted layer.

Value

SpatRaster prediction from model

Examples

```
## Not run:
  m <- glm(z ~ cov1, cov2, data = sdm_data)

  prediction <- predict_sdm(m, covs)


## End(Not run)
```

print.long_tibble *Print method for class* long_tibble

Description

Prints data tables stored in sdmtools for their entire length*, but maintains other nice print features of tbl_df, i.e., tibbles.

Usage

```
## S3 method for class 'long_tibble'
print(x, ...)
```

Arguments

x	An object of class long_tibble.
...	extra arguments for printing

Details

*"entire length" — well really, up to 999 lines, which none currently are.

Examples

```
print(raster_to_terra)
```

raster_to_terra raster *to* terra *equivalence table*

Description

A table of equivalent functions (or near equivalent) from the raster and terra packages.

Usage

```
raster_to_terra
```

Format

global_regions **A tibble with 42 rows and 3 columns::**

raster function name in raster

terra function name in terra

comment Annotations about differences in arguments or usage, etc.

Details

Some argument names may differ.

Source

Initially produced by Gerry Ryan and supplemented with "New method names" section in https://cran.r-project.org/web/packages/terra/terra.pdf, annotations added by IDEM members

---

`rastpointplot` *Raster plot with points*

Description

Simple convenience function to plot points over a raster. Useful for quick data checks.

Usage

```
rastpointplot(r, v, pch = 16, cex = 0.5)
```

Arguments

`r`	`spatRaster` object
`v`	`spatVector` object
`pch`	`integer` point symbol. See ?par
`cex`	`numeric` point size multiplier. See `?par`

Value

graphical plot

Examples

```
r <- example_raster()
v <- example_vector()
rastpointplot(r,v)
```

save_plot *Save plot*

Description

Single line wrapper to save plot as png

Usage

```
save_plot(
  p,
  filename,
  width = 2400,
  height,
  units = c("px", "cm", "mm"),
  res = 300
)
```

Arguments

p	A plot object
filename	character to save plot
width	In units
height	In units. If missing, height will be scaled to width at the golden ratio.
units	Units of width and height. "px" — pixels, the default, "cm", or "mm". "in" are not allowed on principle.
res	resolution, default 300px

Value

nothing. Writes a plot to filename

Examples

```
## Not run:

lovely_plot <- plot(1:10)
# why use three lines
png("lovely_plot.png")
lovely_plot
dev.off()

#when you could use one
save_plot(p = lovely_plot, "lovely_plot.png")


## End(Not run)
```

set_layer_names *set layer names*

Description

Convenience function for setting SpatRaster layer names, enables setting in piped workflows.

Usage

```
set_layer_names(x, layernames)
```

Arguments

x	SpatRaster
layernames	character of length nlyr(x)

Value

SpatRaster

Examples

```
example_raster() |>
 set_layer_names("Charlie Watts")
```

set_levels *set levels*

Description

Convenience function for setting SpatRaster levels names to enable setting in piped workflows.

Usage

```
set_levels(x, levs)
```

Arguments

x	SpatRaster
levs	data.frame

Value

SpatRaster

Examples

```
## Not run:
categorical_raster |>
 set_levels(
  levs = tribble(
    ~value, ~category,
    30, "URBAN CENTRE",
    23, "DENSE URBAN CLUSTER",
    22, "SEMI-DENSE URBAN CLUSTER",
    21, "SUBURBAN OR PERI-URBAN",
    13, "RURAL CLUSTER",
    12, "LOW DENSITY RURAL",
    11, "VERY LOW DENSITY RURAL",
    10, "WATER"
   )
  ) %>%
    as.data.frame()

## End(Not run)
```

---

source_R *Source all R scripts in a directory*

Description

Sources all scripts in a specified directory and optionally prints paths.

Usage

```
source_R(path = "R", print.names = TRUE)
```

Arguments

path	Path of directory; default is /R.
print.names	Print path and name of sourced scripts. Default TRUE.

Value

Scripts are sourced to local environment. NULL returned.

Examples

```
## Not run:
  source_R("/Users/frankenstein/project/R")

## End(Not run)
```

split_rast *Split rasters*

Description

Split rasters

Usage

```
split_rast(x, grain = 4, write_temp = FALSE)
```

Arguments

`x`	A `SpatRaster`
`grain`	Grain of splitting.
`write_temp`	write to a tempfile. Default is FALSE

Details

Splits a `SpatRaster` up into a `grain^2` list of approximately equal geographic sized rasters covering the extent of `x`

Value

A list of `SpatRasters` of length `grain^2`.

Examples

```
# Split a raster into four
library(terra)
r <- example_raster()
s <- split_rast(r, grain = 2)
s

# plot with original
 ps <- lapply(
   s,
   FUN = extend,
   y = r
 ) |>
   rast()

 c(r, ps) |>  plot()
```

standardise_rast *Standardise Raster*

Description

Defunct.

DO NOT USE THIS FUNCTION; use `terra::scale` instead

Usage

```
standardise_rast(x)
```

Arguments

x	A `SpatRaster` object.

Value

A `SpatRaster` objects with mean of zero and standard deviation of one.

Examples

```
## Not run:
r <- example_raster()

standardise_rast(r)

## End(Not run)
```

std_rast *Standardise raster*

Description

Standardises all layers in a `SpatRaster` to a scale of 0-1, by dividing by the maximum value in each layer. *Only operates by layer*

Usage

```
std_rast(x, reverse, filename = NULL, overwrite = TRUE)
```

Arguments

x	`SpatRaster` to standardise
reverse	`logical` if `TRUE` will subtract standardised values from 1
filename	Optional `character` path and filename to write output
overwrite	`logical` if `TRUE` will overwrite `filename`

Details

Will break for non-numeric rasters

Value

A `SpatRaster` with range 0-1

Examples

```
example_raster(seed = 3010) |>
  std_rast(reverse = TRUE)
```

`temptif` *temptif*

Description

Returns a temporary file with a `*.tif` extenstion

Usage

```
temptif()
```

Value

`character`

Examples

```
temptif()
```

`writereadrast` *Write and read back* `spatRaster`

Description

When a new `terra::SpatRaster` is created it is stored in memory. Writing it to disc and reading the object back from file requires two steps: `terra::writeRaster` then reading and re-assigning via `terra::rast`. `writereadrast` wraps these into a single step.

Usage

```
writereadrast(x, filename, overwrite = TRUE, layernames = NULL)
```

Arguments

`x`	A `terra::SpatRaster`
`filename`	A `character` file path and name to save `x` to disc.
`overwrite`	`logical`; overwrite existing raster. **NB:** by default, `overwrite` = `TRUE`, this is the opposite of the default behaviour of `terra::writeRaster`
`layernames`	`character` of length `nlyr(x)`

Value

A `terra::SpatRaster` object reading from disc at `filename`.

Examples

```
## Not run:
# create raster then assign
r <- sdmtools::example_raster()

# usual workflow in two slow tedious boring steps
terra::writeRaster(r, "LowerSpringvale.tif")
s <- terra::rast("LowerSpringvale.tif")

Better workflow in one fast enjoyable lithesome step with `sdmtools`
r <- writereadrast(r, "tootgarook.tif")

# or roll into single step with pipe
q <- sdmtools::example_raster() |>
  writereadrast("frankstonfreeway.tif")

## End(Not run)
```

Index

∗ **datasets**
    global_regions, 6
    raster_to_terra, 17

assign_nearest_land, 2

example_mask, 3
example_raster, 3
example_vector, 4
extract_covariates, 4

global_regions, 6

import_rasts, 7
inside_mask, 8

make_africa_mask, 8
make_mpp_list, 9
mask_all, 11
mask_from_all, 12
maskpointsdf, 10
match_ref, 13

nearest_land, 13
new_long_tibble, 14

predict_mpp_rast, 14
predict_mpp_rast_all, 15
predict_sdm, 16
print.long_tibble, 17

raster_to_terra, 17
rastpointplot, 18

save_plot, 19
set_layer_names, 20
set_levels, 20
source_R, 21
split_rast, 22
standardise_rast, 23
std_rast, 23

temptif, 24
terra::extract(), *13*

writereadrast, 24