Package ‘paisaje’

October 21, 2025

Type Package

Title Spatial and Environmental Data Tools for Landscape Ecology

Version 0.1.1

Maintainer Manuel Spínola <mspinola10@gmail.com>

Description Provides functions for landscape analysis and data retrieval.
The package allows users to download environmental variables from global datasets (e.g., WorldClim, ESA WorldCover, Nighttime Lights), and to compute spatial and landscape metrics using a hexagonal grid system based on the H3 spatial index. It is useful for ecological modeling, biodiversity studies, and spatial data processing in landscape ecology.
Fick and Hijmans (2017) <doi:10.1002/joc.5086>.
Zanaga et al. (2022) <doi:10.5281/zenodo.7254221>.
Uber Technologies Inc. (2022) ``H3: Hexagonal hierarchical spatial index''.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

URL https://manuelspinola.github.io/paisaje/,
https://github.com/ManuelSpinola/paisaje

BugReports https://github.com/ManuelSpinola/paisaje/issues

Imports rlang, dplyr, tidyr, sf, terra, h3jsr, rvest, httr, landscapemetrics, spocc, exactextractr, progress, magrittr

Suggests ggplot2, rmarkdown, knitr

VignetteBuilder knitr

Depends R (>= 4.1.0)

LazyData true

Language en-US

NeedsCompilation no

Author Manuel Spínola [aut, cre] (ORCID: <https://orcid.org/0000-0002-7839-1908>)

Repository CRAN

Date/Publication 2025-10-21 08:20:02 UTC

Contents

`calculate_it_metrics` *Calculate 5 information theory landscape metrics*

Description

This function allow to calculate 5 information theory landscape metrics

Usage

```
calculate_it_metrics(landscape_raster, aoi_sf)
```

Arguments

`landscape_raster`	A categorical raster object: SpatRaster.
`aoi_sf`	The spatial area of interest as an sf object.

Details

Calculate the landscape metrics: condent, ent, joinent, mutinf, and relmutinf.

Value

An sf object

Note

This is a wrapper of the function "sample_lsm" from the landscapemetrics package (see References)

References

Hesselbarth, M.H.K., Sciaini, M., With, K.A., Wiegand, K., Nowosad, J. 2019. landscapemetrics: an open-source R tool to calculate landscape metrics. Ecography, 42: 1648-1657 (v2.1.4).

Nowosad J., TF Stepinski. 2019. Information theory as a consistent framework for quantification and classification of landscape patterns. https://doi.org/10.1007/s10980-019-00830-x

Information theory-based framework for the analysis of landscape patterns

Examples

```
nc <- sf::st_read(system.file("shape/nc.shp", package = "sf"))
nc <- sf::st_transform(nc, crs = 4326)

clc <- terra::rast(system.file("sao_miguel/clc2018_v2020_20u1.tif",
  package = "exactextractr"))

bbox <- sf::st_bbox(clc) |>
  sf::st_as_sfc() |>
  sf::st_as_sf()

h3_bbox <- paisaje::get_h3_grid(bbox, resolution = 6)

result_sf <- paisaje::calculate_it_metrics(clc, h3_bbox)
```

`count_points_in_polygons`

Count Points within Polygons by Species

Description

Counts the number of points per species within each polygon. If the points dataset contains a ‘species‘ column, a separate column is created for each species with the counts inside each polygon. Spaces in species names are replaced with underscores for naming columns.

This function is particularly useful in ecological studies where species have different spatial distributions. It accounts for the possibility that some species may not be present in all polygons, producing zero counts in those cases.

Usage

```
count_points_in_polygons(points_sf, polygons_sf)
```

Arguments

`points_sf`	An ‘sf‘ object containing point geometries. Must include a ‘species‘ column.
`polygons_sf`	An ‘sf‘ object containing polygon geometries.

Details

The function performs a spatial join to count occurrences of each species within each polygon. For species absent in a polygon, the count will be zero. This approach allows for flexible analysis of species distributions across landscape units.

Value

An ‘sf‘ object containing the original polygons and additional columns for each species count. Column names follow the format ‘species_name_count‘, with spaces replaced by underscores.

Examples

```
library(sf)

points_sf <- st_as_sf(data.frame(
  id = 1:6,
  species = c("Panthera onca", "Panthera onca", "Felis catus",
              "Felis catus", "Felis catus", "Panthera leo"),
  x = c(1, 2, 3, 4, 5, 6),
  y = c(1, 2, 3, 4, 5, 6)
), coords = c("x", "y"), crs = 4326)

polygons_sf <- st_as_sf(data.frame(
  id = 1:2,
  geometry = st_sfc(
    st_polygon(list(rbind(c(0,0), c(3,0), c(3,3), c(0,3), c(0,0)))),
    st_polygon(list(rbind(c(3,3), c(6,3), c(6,6), c(3,6), c(3,3))))
  )
), crs = 4326)

result <- count_points_in_polygons(points_sf, polygons_sf)
print(result)
```

create_cat_esa_10m *Create Categorical Land Cover Raster from Copernicus ESA WorldCover Data*

Description

This function takes a ‘SpatRaster‘ object containing Copernicus ESA WorldCover land cover data, reclassifies it into categorical land cover classes based on predefined schemes, and returns the resulting categorical raster.

Usage

```
create_cat_esa_10m(land_cover)
```

Arguments

`land_cover`	A ‘SpatRaster‘ object representing the input land cover raster from Copernicus ESA WorldCover. This raster should contain land cover classes as defined by the Copernicus program.

Details

The function uses a predefined classification scheme for ESA WorldCover data, assigning numeric or categorical values to represent different land cover types. The resulting raster can be used for further spatial analysis or landscape ecology studies.

Value

A ‘SpatRaster‘ object containing the reclassified categorical land cover raster. Each pixel will have a category corresponding to a defined land cover type.

References

Zanaga, D., Van De Kerchove, R., De Keersmaecker, W., et al. (2021). ESA WorldCover 10 m 2020 v100. https://doi.org/10.5281/zenodo.5571936 Zanaga, D., Van De Kerchove, R., Daems, D., et al. (2022). ESA WorldCover 10 m 2021 v200. https://doi.org/10.5281/zenodo.7254221 ESA WorldCover project 2020 and 2021. Contains modified Copernicus Sentinel data processed by ESA WorldCover consortium. ESA WorldCover

Examples

```
## Not run:
# Assuming 'land_cover_raster' is a SpatRaster object from ESA WorldCover
cat_raster <- create_cat_esa_10m(land_cover_raster)

## End(Not run)
```

`cr_outline_c` *Costa Rica Continental Outline*

Description

A simplified outline of Costa Rica as an ‘sf‘ object.

Usage

```
cr_outline_c
```

Format

An ‘sf‘ object containing polygon geometry of Costa Rica.

Source

Adapted from publicly available geographic data.

Examples

```
library(sf)
plot(cr_outline_c)
```

extract_cat_raster *Extract categorical raster values by polygons or hexagons*

Description

Extracts categorical raster values (e.g., land cover classes) for each polygon in a given grid (e.g., H3 hexagons or administrative units). It returns either the proportion of each category within each polygon or the raw counts.

Usage

```
extract_cat_raster(spat_raster_cat, sf_hex_grid, proportion = TRUE)
```

Arguments

`spat_raster_cat`	A categorical raster of class ‘SpatRaster‘ from the **terra** package.
`sf_hex_grid`	An object of class ‘sf‘ representing the polygons (e.g., hexagonal grid).
`proportion`	Logical. If ‘TRUE‘ (default), returns the proportion of each category within each polygon. If ‘FALSE‘, returns counts instead.

Details

This function uses ‘exactextractr::exact_extract()‘ to accurately compute the overlap between polygons and raster cells. Invalid geometries are automatically fixed using ‘sf::st_make_valid()‘, and only ‘POLYGON‘ or ‘MULTIPOLYGON‘ geometries are processed.

Value

An ‘sf‘ object with the extracted values joined to the input grid. Each polygon will contain the calculated proportions or counts of the categorical raster values that fall within its area.

Examples

```
## Not run:

# Example categorical raster
r <- terra::rast(nrows = 10, ncols = 10)
terra::values(r) <- sample(1:3, terra::ncell(r), replace = TRUE)

# Example grid (hexagons)
bbox <- sf::st_as_sfc(sf::st_bbox(terra::as.polygons(r)))
hex <- sf::st_make_grid(bbox, cellsize = 0.2, square = FALSE)
hex_sf <- sf::st_sf(ID = 1:length(hex), geometry = hex)

# Extract proportions of land cover classes per hexagon
result <- extract_cat_raster(r, hex_sf, proportion = TRUE)

## End(Not run)
```

`extract_num_raster` *Extract Numeric Raster Values by Polygons*

Description

Extracts numeric raster values for each polygon in an sf object. Uses exactextractr to compute the weighted mean using the area of overlap.

Usage

```
extract_num_raster(spat_raster_multi, sf_hex_grid)
```

Arguments

`spat_raster_multi`	A SpatRaster object (single or multilayer numeric raster).
`sf_hex_grid`	An sf object with polygon geometries (e.g., H3 hexagons).

Value

An sf object with additional columns for each raster layer.

Examples

```
## Not run:

r <- terra::rast(system.file("raster/bio.tif", package = "spData"))
grid_sf <- get_h3_grid(st_as_sf(st_bbox(r)), resolution = 6)
result_sf <- extract_num_raster(r, grid_sf)
head(result_sf)
```

```
## End(Not run)
```

get_esa_10m *Download ESA WorldCover land cover data*

Description

Downloads ESA WorldCover land cover data at 10 m resolution for a specified area of interest (AOI) and year. Useful for landscape ecology studies, environmental analyses, and habitat mapping.

Usage

```
get_esa_10m(aoi_sf, year = 2020, output_folder = NULL)
```

Arguments

`aoi_sf` ‘sf‘ An sf object defining the area of interest (AOI). This can be a country, state, or custom boundary.

`year` ‘numeric‘ Year of the land cover data. Available: - 2020: ESA WorldCover 10 m 2020 v100 - 2021: ESA WorldCover 10 m 2021 v200

`output_folder` ‘character‘ Directory where data files will be saved. Default is "." (current working directory).

Details

This function downloads global land cover raster data produced by the ESA WorldCover project. The downloaded file can be large (hundreds of MB), and processing may take several minutes depending on the AOI size and internet connection speed.

Value

‘SpatRaster‘ A raster object containing land cover classification for the specified AOI and year. The raster values correspond to land cover classes as defined by the ESA WorldCover classification scheme.

References

Zanaga, D., Van De Kerchove, R., De Keersmaecker, W., et al. (2021). ESA WorldCover 10 m 2020 v100. https://doi.org/10.5281/zenodo.5571936 Zanaga, D., Van De Kerchove, R., Daems, D., et al. (2022). ESA WorldCover 10 m 2021 v200. https://doi.org/10.5281/zenodo.7254221

Examples

```
library(sf)
nc <- st_read(system.file("shape/nc.shp", package="sf"))
get_esa_10m(nc, year = 2021, output_folder = tempdir())
```

`get_h3_grid` *Generate an H3 Hexagonal Grid for an sf Object*

Description

Generates a hexagonal grid of H3 cells at a specified resolution that intersect with a given ‘sf‘ object. This is a wrapper for functions from the **h3jsr** package.

Usage

```
get_h3_grid(sf_object, resolution = 6, expand_factor = 0.1)
```

Arguments

`sf_object`	(`sf`) An sf object defining the area of interest. Must have a valid coordinate reference system (CRS).
`resolution`	(`integer`) H3 resolution level (1–16). Default is 6. Lower values produce fewer, larger hexagons (faster processing, coarser grid).
`expand_factor`	(`numeric`) Expands bounding box to ensure coverage. Default is 0.1.

Details

Reducing the resolution (e.g., 5 or 6) can greatly reduce processing time and memory usage, especially for large AOIs. Each decrease in resolution increases the size of individual hexagons exponentially.

Value

(`sf`) An sf object containing the hexagonal grid polygons covering the input area. Each polygon represents an H3 cell at the specified resolution, with a column containing the H3 index.

Examples

```
library(sf)
nc <- st_read(system.file("shape/nc.shp", package="sf"))
h3_grid_sf <- get_h3_grid(nc, resolution = 6)
```

get_nightlight_data *Download and Retrieve Nightlight Data*

Description

Downloads nightlight data from the Earth Observation Group's website. It scrapes the website to locate and download the latest available nightlight dataset for the specified year and month.

Usage

```
get_nightlight_data(
  year,
  month,
  version = "v10",
  destination_dir = NULL,
  timeout = 1200
)
```

Arguments

`year`	'numeric' o 'character' The year for which to download nightlight data (e.g., 2020).
`month`	'numeric' o 'character' Month of the year (1–12). Will be formatted as two digits (e.g., '"03"' for March).
`version`	'character' Nightlight data version. Default is '"v10"'.
`destination_dir`	'character' Directory where the downloaded '.tif' file will be saved. Default is the current working directory '"."'.
`timeout`	'numeric' Timeout in seconds for the download. Default is '1200' seconds.

Details

The function constructs the appropriate URL for the specified year, month, and data version, then scrapes the directory listing to locate the latest available '.tif' file. It downloads and saves the file to the 'destination_dir'. This function is useful for retrieving nightlight data for studies involving human activity, urbanization, and environmental monitoring.

Value

'character' o 'NULL' Path to the downloaded '.tif' file. Returns 'NULL' if no file was found or if an error occurred.

Examples

```
# Download nightlight data for March 2021
file_path <- get_nightlight_data(2021, 3)
print(file_path)
```

`get_records` *Retrieve species occurrence records within an Area of Interest*

Description

Retrieves species occurrence records from specified data providers within a given Area of Interest (AOI). The results are returned as an ‘sf‘ object containing point geometries. Duplicates can be removed based on geometry. This is a wrapper of the ‘occ‘ function from the spocc package.

Usage

```
get_records(species, aoi_sf, providers = NULL,
  limit = 500, remove_duplicates = FALSE, date = NULL)
```

Arguments

`species`	(‘character‘) Vector of species names to query.
`aoi_sf`	(‘sf‘) An ‘sf‘ object representing the Area of Interest. Must have a valid CRS.
`providers`	(‘character‘) Data providers to query (e.g., ‘"gbif"‘, ‘"inat"‘). Default is ‘NULL‘, which queries all available providers.
`limit`	(‘integer‘) Maximum number of records to retrieve per provider. Default is 500.
`remove_duplicates`	(‘logical‘) Whether to remove duplicate geometries. Default is ‘FALSE‘.
`date`	(‘character‘) Vector of length two specifying the date range (e.g., ‘c("YYYY-MM-DD", "YYYY-MM-DD")‘). Records outside this range will be excluded. Default is ‘NULL‘ (no date filtering).

Details

This function simplifies retrieving occurrence records by wrapping the functionality of the ‘spocc::occ‘ function. It handles AOI spatial filtering and optional removal of duplicates.

Value

(‘sf‘) An ‘sf‘ object containing species occurrence records within the specified AOI that match the query criteria. Returns ‘NULL‘ if no records are found or if input parameters are invalid.

Examples

```
library(sf)
nc <- sf::st_read(system.file("shape/nc.shp", package="sf"))
records <- get_records(
  species = "Lynx rufus",
  aoi_sf = nc,
  providers = c("gbif", "inat"),
  limit = 200
)
head(records)
```

get_records_by_hexagon

Retrieve species records aggregated by H3 hexagons

Description

Downloads species occurrence data within a specified Area of Interest (AOI) and aggregates these records into H3 hexagonal grid cells at a given resolution. Returns an ‘sf‘ object with one polygon per hexagon and columns containing species occurrence counts.

Usage

```
get_records_by_hexagon(
  species, aoi_sf, res = 6,
  providers = NULL, remove_duplicates = FALSE,
  date = NULL, expand_factor = 0.1, limit = 500
)
```

Arguments

species	character vector. Species names to query.
aoi_sf	sf object. Area of Interest polygon.
res	integer. H3 resolution level (1–16). Default: 6.
providers	character vector. Data providers to query. Default: NULL (all).
remove_duplicates	logical. Remove duplicate records. Default: FALSE.
date	character vector of length two. Start and end dates for filtering records.
expand_factor	numeric. Expand AOI bounding box. Default: 0.1.
limit	integer. Maximum number of occurrence records per species. Default: 500.

Details

This function is useful for spatial biodiversity analyses where data should be aggregated into a uniform spatial grid. The H3 grid system enables multi-resolution analysis and efficient spatial summarization of point occurrence data.

Value

sf object. H3 hex grid with species occurrence counts.

Examples

```
library(sf)
nc <- sf::st_read(system.file("shape/nc.shp", package="sf"))
hex_counts <- get_records_by_hexagon(
  species = c("Lynx rufus"),
  aoi_sf = nc,
  res = 6,
  limit = 200
)
print(hex_counts)
```

---

get_worldclim_future *Download and process future environmental variables from WorldClim v2.1*

Description

Downloads future climate data from WorldClim based on CMIP6 climate models and SSP scenarios. The data can be retrieved at various spatial resolutions and optionally clipped to a specified area of interest (AOI).

Usage

```
get_worldclim_future(
  var = "bioc",
  res = "30s",
  scenario = "585",
  time_range = "2021-2040",
  gcm = "ACCESS-CM2",
  aoi = NULL,
  retries = 3,
  timeout = 300,
  destination_dir = NULL
)
```

Arguments

`var` character Climate variable to download. Options:

- "bioc" — Bioclimatic variables (19 variables)
- "prec" — Precipitation
- "tavg" — Average temperature
- "tmin" — Minimum temperature
- "tmax" — Maximum temperature

Default is "bioc".

`res` character Spatial resolution of the data. Options:

- "30s" — ~1 km (30 arc-seconds)
- "2.5m" — ~5 km (2.5 arc-minutes)
- "5m" — ~10 km (5 arc-minutes)
- "10m" — ~20 km (10 arc-minutes)

Default is "30s".

`scenario` character SSP scenario. Options:

- "126" — SSP1-2.6 (low emissions)
- "245" — SSP2-4.5 (intermediate emissions)
- "370" — SSP3-7.0 (high emissions)
- "585" — SSP5-8.5 (very high emissions)

Default is "585".

`time_range` character Time period. Options:

- "2021-2040"
- "2041-2060"
- "2061-2080"
- "2081-2100"

Default is "2021-2040".

`gcm` character General Circulation Model. Options: "ACCESS-CM2", "ACCESS-ESM1-5", "AWI-CM-1-1-MR", "BCC-CSM2-MR", "CanESM5", "CanESM5-CanOE", "CMCC-ESM2", "CNRM-CM6-1", "CNRM-CM6-1-HR", "CNRM-ESM2-1", "EC-Earth3-Veg", "EC-Earth3-Veg-LR", "FIO-ESM-2-0", "GFDL-ESM4", "GISS-E2-1-G", "GISS-E2-1-H", "HadGEM3-GC31-LL", "INM-CM4-8", "INM-CM5-0", "IPSL-CM6A-LR", "MIROC-ES2L", "MIROC6", "MPI-ESM1-2-HR", "MPI-ESM1-2-LR", "MRI-ESM2-0", "UKESM1-0-LL". Default is "ACCESS-CM2".

`aoi` sf or SpatRaster Optional area of interest to clip the data. Default is NULL (no clipping).

`retries` integer Number of attempts to retry download in case of failure. Default is 3.

`timeout` numeric Download timeout in seconds. Default is 300.

`destination_dir`
character Directory where downloaded data will be stored. Default is NULL (uses a temporary directory).

Value

SpatRaster object containing the selected climate variables, optionally clipped to the specified AOI.

References

Fick, S. E., & Hijmans, R. J. (2017). WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas. International Journal of Climatology, 37(12), 4302–4315. doi:10.1002/joc.5086

Examples

```
nc <- sf::st_read(system.file("shape/nc.shp", package = "sf"))
nc <- sf::st_transform(nc, crs = 4326)

climate_future <- paisaje::get_worldclim_future(
  var = "tmin", res = "10m", scenario = "585",
  time_range = "2021-2040", gcm = "ACCESS-CM2", aoi = nc
)
```

---

get_worldclim_historic

Descargar y procesar variables climáticas históricas de WorldClim v2.1

Description

Descarga datos climáticos históricos de WorldClim v2.1 y los procesa según los parámetros especificados. Soporta múltiples variables climáticas y resoluciones espaciales. Opcionalmente recorta los datos a un área de interés (AOI).

Usage

```
get_worldclim_historic(
  var = "bio",
  res = 10,
  aoi = NULL,
  retries = 3,
  timeout = 300,
  destination_dir = NULL
)
```

Arguments

`var` Character. Variable climática a descargar. Opciones:

- "bio" — Variables bioclimáticas.
- "tavg" — Temperatura media.

- "tmin" — Temperatura mínima.
- "tmax" — Temperatura máxima.
- "prec" — Precipitación.
- "srad" — Radiación solar.
- "wind" — Velocidad del viento.
- "vapr" — Presión de vapor.

Por defecto: ‘"bio"‘.

`res` Numeric. Resolución espacial en minutos de arco. Valores válidos: ‘0.5‘, ‘2.5‘, ‘5‘, ‘10‘. Estos valores se mapean internamente a cadenas aceptadas por WorldClim:

- 0.5 → "30s"
- 2.5 → "2.5m"
- 5 → "5m"
- 10 → "10m"

Por defecto: ‘10‘.

`aoi` sf o SpatRaster opcional. Área de interés para recortar los datos.

`retries` Integer. Número de intentos de descarga en caso de fallo. Por defecto: ‘3‘.

`timeout` Numeric. Tiempo máximo de descarga en segundos. Por defecto: ‘300‘.

`destination_dir`

Character. Carpeta donde guardar los datos descargados. Si NULL, se usa un directorio temporal.

Value

Un objeto ‘SpatRaster‘ con las variables climáticas históricas. Si se especifica ‘aoi‘, los datos se recortan a esa área.

References

Fick, S. E., & Hijmans, R. J. (2017). WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas. International Journal of Climatology, 37(12), 4302–4315. doi:10.1002/joc.5086

Examples

```
nc <- sf::st_read(system.file("shape/nc.shp", package="sf"))
nc <- sf::st_transform(nc, crs = 4326)

climate_historic <- get_worldclim_historic(
  var = "tmin",
  res = 5,
  aoi = nc
)
```

Index