Package: paisaje (via r-universe)

June 13, 2026

Type Package

Title Spatial and Environmental Data Tools for Landscape Ecology

Version 0.3.0

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, CHELSA, 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''. Román et al. (2018) <doi:10.1016/j.rse.2018.03.017>. Karger et al. (2017) <doi:10.1038/sdata.2017.122>. Brun et al. (2022) <doi:10.5194/essd-14-5573-2022>.

License MIT + file LICENSE

Encoding UTF-8

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

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

Imports dplyr, tidyr, sf, terra, h3jsr, httr, landscapemetrics, spocc, exactextractr, progress, blackmarbler, geodata

Suggests ggplot2, rmarkdown, knitr

VignetteBuilder knitr

Depends R (>= 4.1.0)

LazyData true

Language en-US

Config/roxygen2/version 8.0.0

NeedsCompilation no

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

Repository https://cran.r-universe.dev

Date/Publication 2026-06-12 09:17:02 UTC

RemoteUrl https://github.com/cran/paisaje

RemoteRef HEAD

RemoteSha e71750952a9498c1d554b5116b9d43cc85dc3394

Contents

`calculate_it_metrics` *Calculate Landscape Complexity Metrics (IT Metrics) per Polygon*

Description

Calculates specified landscape complexity metrics (a subset of Information Theory metrics) from a categorical land-cover raster for each input polygon using `landscapemetrics::sample_lsm()`. This function ensures a safe, alignment-guaranteed join of the results back to the original geometry.

Usage

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

Arguments

`landscape_raster`
: A `SpatRaster` object representing the categorical landscape (e.g., LULC).

`aoi_sf`
: An `sf` object containing polygonal geometries (e.g., H3 hexagons) for which the landscape metrics should be calculated.

Details

This function calculates metrics at the `"landscape"` level, filtering for `"complexity metric"` types. The function prioritizes data integrity by adding a temporary `plot_id` column based on row index, which is used by `landscapemetrics`.

Crucially, the function uses `dplyr::left_join` with this `plot_id` for merging the results. This **robust join method** prevents data misalignment that could occur if rows were dropped during metric calculation, which is a significant improvement over the unsafe `cbind` method. The temporary `plot_id` column is removed before the final object is returned.

Value

An `sf` object identical to `aoi_sf`, but with new columns appended. The new columns represent the calculated landscape metrics (e.g., `lsm_shdi`) with an `lsm_` prefix.

See Also

`sample_lsm` for available metrics.

Examples

```
## Not run:
# Assuming 'lulc' is a SpatRaster and 'hex_grid_sf' is an sf polygon grid
# metrics_sf <- calculate_it_metrics(lulc, hex_grid_sf)
# head(metrics_sf)

## End(Not run)
```

`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)
```

`cr_outline` *Costa Rica Full Outline (Continental + Islands)*

Description

An `sf` multipolygon containing the full outline of Costa Rica, derived from GADM 4.1. Includes the continental landmass, the Isla del Coco (~550 km offshore in the Pacific Ocean), and all other minor oceanic islands.

For the continental outline only (without islands), see `cr_outline_c`.

Usage

```
cr_outline
```

Format

An `sf` object with 1 feature and 1 column:

geometry MULTIPOLYGON in WGS 84 (EPSG:4326) representing the full national territory of Costa Rica including all islands.

Details

When to use this vs cr_outline_c Use `cr_outline` when your analysis requires the full national territory (e.g., legal/administrative boundaries, marine protected areas, or studies specifically involving the Isla del Coco). Use `cr_outline_c` for mainland ecological analyses where oceanic islands would distort results (species distribution models, landscape metrics, climate extraction).

Reproducibility Generated with `data-raw/cr_outline.R`. To regenerate:

```
source("data-raw/cr_outline.R")
```

Source

Global Administrative Areas (GADM) version 4.1. Downloaded via `geodata::gadm("CRI", level = 0)`. https://gadm.org

See Also

- `cr_outline_c` — continental outline only (no islands).
- `get_cr_outline` — programmatic version with options.
- `get_h3_grid` — generate H3 hexagonal grid over this AOI.

Examples

```
data(cr_outline)
plot(sf::st_geometry(cr_outline), main = "Costa Rica (full territory)")

## Not run:
# Compare continental vs full
par(mfrow = c(1, 2))
plot(sf::st_geometry(cr_outline_c), main = "Continental")
plot(sf::st_geometry(cr_outline),   main = "Full territory")

## End(Not run)
```

---

cr_outline_c *Costa Rica Continental Outline*

Description

An `sf` polygon containing the continental outline of Costa Rica, derived from GADM 4.1. The Isla del Coco and all other minor oceanic islands have been removed, retaining only the largest polygon (the continental landmass).

For the full outline including all islands, see `cr_outline`.

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

Usage

```
cr_outline_c

cr_outline_c
```

Format

An `sf` object with 1 feature and 1 column:

geometry POLYGON in WGS 84 (EPSG:4326) with 30,261 vertices, representing the continental outline of Costa Rica.

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

Details

Island removal Costa Rica includes the Isla del Coco (~550 km offshore in the Pacific), which is excluded here. The continental polygon is obtained by casting the GADM multipolygon to individual polygons and retaining the one with the largest area. For analyses requiring all national territory, use `cr_outline`.

Reproducibility Generated with `data-raw/cr_outline.R`. To regenerate:

```
source("data-raw/cr_outline.R")
```

Source

Global Administrative Areas (GADM) version 4.1. Downloaded via `geodata::gadm("CRI", level = 0)`. https://gadm.org

Adapted from publicly available geographic data.

See Also

- `cr_outline` — full outline including all islands.
- `get_cr_outline` — programmatic version with options.
- `get_h3_grid` — generate H3 hexagonal grid over this AOI.

Examples

```
data(cr_outline_c)
plot(sf::st_geometry(cr_outline_c), main = "Costa Rica (continental)")

## Not run:
bio1 <- get_chelsa_historic(var = "bio1", aoi = cr_outline_c)

## End(Not run)

library(sf)
plot(cr_outline_c)
```

`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)
```

---

extract_cat_raster — *Calculate Area Proportions for Categorical Raster Classes (Generic)*

Description

Extracts and calculates the **area proportion** of each categorical class (e.g., LULC) found within each input polygon. This function uses area-weighting to ensure highly accurate, sub-pixel zonal statistics.

Usage

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

Arguments

`spat_raster_cat`
: A single-layer `SpatRaster` object containing categorical values.

`sf_hex_grid`
: An `sf` object containing polygonal geometries. The function will use `h3_address` if present, otherwise it creates and uses a temporary `ID` column for joining.

`proportion`
: Logical. If `TRUE` (default), the output values are the proportion of the polygon area covered by each category (summing to 1 for the covered area). If `FALSE`, the output is the raw sum of the coverage fraction (area).

Details

This function replaces the simplistic, non-area-weighted `table()` counting method with a robust custom function utilizing `dplyr` and the `coverage_fraction` column from `exactextractr`. Key features include:

- **Area-Weighted Accuracy:** Uses `coverage_fraction` for precise results.
- **NA Filtering:** Excludes `NA` raster values to prevent a `prop_NaN` column.
- **Numerical Ordering:** Sorts the final output columns by category number (e.g., 70 before 80).

Value

An `sf` object identical to `sf_hex_grid`, but with new columns appended for each categorical value found in the raster. Column names follow the pattern <layer_name>_prop_<category_value>. Columns are **numerically ordered** by the category value.

Examples

```
## Not run:
# Assuming 'lulc' is a categorical SpatRaster and 'hex_grid' is an sf polygon grid
# cat_data_p <- extract_cat_raster(lulc, hex_grid)
# head(cat_data_p)

## End(Not run)
```

`extract_num_raster` *Extract Area-Weighted Mean from Numeric Raster Stack for Polygons*

Description

Calculates the area-weighted mean value for each layer in a numeric `SpatRaster` (or single layer) within each polygon feature of an `sf` object. This function is designed for high-precision zonal statistics of continuous variables (e.g., bioclimatic data).

Usage

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

Arguments

`spat_raster_multi`
: A `SpatRaster` object from the `terra` package. Must contain numeric layers (can be a single layer or a stack/brick).

`sf_hex_grid`
: An `sf` object containing polygonal geometries (e.g., H3 hexagons).

Details

The function uses `exactextractr::exact_extract` with `fun = "weighted_mean"` and `weights = "area"` to ensure the most accurate sub-pixel summary. A critical security check is implemented before binding columns (`bind_cols`) to prevent data misalignment in case of row count discrepancies between the input features and the extracted results.

Value

An `sf` object identical to `sf_hex_grid`, but with new columns appended. The new column names match the original `SpatRaster` layer names. The values represent the area-weighted mean for that variable within each polygon.

Examples

```
## Not run:
# Assuming 'bio' is a SpatRaster stack and 'h7' is an sf hexagon grid
# bio_p <- extract_num_raster(bio, h7)
# head(bio_p)

## End(Not run)
```

---

get_chelsa_future — *Download and Process Future Climate Variables from CHELSA v2.1 (CMIP6)*

Description

Downloads future bioclimatic variables from CHELSA v2.1 under CMIP6 climate scenarios. Data are available for three SSP scenarios, five GCMs (ISIMIP3b selection), and three future periods. Files are served as Cloud Optimized GeoTIFFs (COGs) from the Swiss WSL EnviCloud, enabling efficient spatial subsetting via `/vsicurl/` without downloading global files.

One or more bioclimatic variables (bio1–bio19) can be requested in a single call. The result is a multi-layer `SpatRaster` optionally cropped and masked to the AOI, consistent with the interface of `get_worldclim_future`.

Usage

```
get_chelsa_future(
  var = "bio1",
  scenario = "ssp585",
  period = "2041-2070",
  gcm = "MPI-ESM1-2-HR",
  aoi = NULL,
  destination_dir = NULL,
  timeout = 300
)
```

Arguments

var 'character' vector. One or more CHELSA bioclimatic variable names ("bio1" through "bio19"), or "all" to download all 19. Default: "bio1".

scenario 'character'. SSP emission scenario. Options:

- "ssp126" - SSP1-2.6 (low emissions, sustainable development).
- "ssp370" - SSP3-7.0 (high emissions, regional rivalry).
- "ssp585" - SSP5-8.5 (very high emissions, fossil-fueled growth).

Default: "ssp585".

period 'character'. Future climatological period. Options:

- "2011-2040" - Near future.
- "2041-2070" - Mid future.
- "2071-2100" - Far future.

Default: "2041-2070".

gcm 'character'. Global Circulation Model following the ISIMIP3b selection. Options:

- "GFDL-ESM4" - Priority 1 (highest priority).
- "IPSL-CM6A-LR" - Priority 2.
- "MPI-ESM1-2-HR" - Priority 3.
- "MRI-ESM2-0" - Priority 4.
- "UKESM1-0-LL" - Priority 5.

When fewer than five models are used, selection should follow priority order. Default: "MPI-ESM1-2-HR".

aoi 'sf' or 'SpatVector' or 'NULL'. Area of interest used to crop and mask the raster. If NULL (default), the global raster is returned. Providing an AOI is strongly recommended given file sizes.

destination_dir 'character' or 'NULL'. Directory where the output .tif will be saved. If NULL, a temporary directory is used.

timeout 'numeric'. Maximum time in seconds per HTTP request. Default: 300.

Details

Spatial resolution CHELSA v2.1 future projections are at a **fixed resolution of 30 arc-seconds (~1 km)**. There is no res parameter - unlike WorldClim, CHELSA does not offer coarser resolutions. To downsample, use terra::aggregate() on the returned SpatRaster.

GCM availability CHELSA v2.1 future projections follow the ISIMIP3b model selection, which provides five GCMs covering a range of climate sensitivities and regional performance. Not all SSP x GCM x period combinations are guaranteed to be available on the server. If a combination is unavailable, the function emits a warning and returns NULL for that variable.

SSP scenarios available CHELSA v2.1 provides SSP126, SSP370, and SSP585. Note that SSP245 (available in WorldClim) is **not available** in CHELSA v2.1 future bioclimatic variables.

COG streaming Files use the /vsicurl/ GDAL virtual filesystem to stream only the tiles covering aoi from the remote COG, avoiding full file downloads.

Comparison with get_worldclim_future()

- CHELSA uses mechanistic downscaling; WorldClim uses statistical interpolation.
- CHELSA offers SSP126/370/585; WorldClim offers SSP126/245/370/585.
- CHELSA provides 5 GCMs (ISIMIP3b); WorldClim provides 23 GCMs.
- Both are at ~1 km (30 arc-second) resolution.
- For regions with complex terrain, CHELSA is generally preferred.

Value

A SpatRaster with one layer per requested variable. If aoi is provided, the raster is cropped and masked to it. Also written to destination_dir as a single multi-layer .tif. Returns NULL invisibly on error.

References

Karger, D. N., Conrad, O., B00f6hner, J., Kawohl, T., Kreft, H., Soria-Auza, R. W., Zimmermann, N. E., Linder, P., & Kessler, M. (2017). Climatologies at high resolution for the earth's land surface areas (CHELSA). *Scientific Data*, 4, 170122. doi:10.1038/sdata.2017.122

Brun, P., Zimmermann, N. E., Hari, C., Pellissier, L., & Karger, D. N. (2022). Global climate-related predictors at kilometre resolution for the past and future. *Earth System Science Data*, 14, 5573-5603. doi:10.5194/essd1455732022

See Also

- get_chelsa_historic - CHELSA 1981-2010 baseline.
- get_worldclim_future - WorldClim v2.1 future projections.
- extract_num_raster - extract area-weighted means per polygon.
- CHELSA CMIP6: https://chelsa-climate.org/
- EnviCloud browser: https://envicloud.wsl.ch/#/?bucket=https://os.zhdk.cloud.switch.ch/chelsav2/

Examples

```
## Not run:
library(sf)

# Use Costa Rica outline (included in paisaje)
aoi <- paisaje::cr_outline_c

# Single variable - Annual mean temperature, mid-century, pessimistic
bio1_fut <- get_chelsa_future(
  var      = "bio1",
  scenario = "ssp585",
  period   = "2041-2070",
  gcm      = "MPI-ESM1-2-HR",
```

```
  aoi      = aoi
)

# Multiple variables - near future, optimistic
bio_stack <- get_chelsa_future(
  var      = c("bio1", "bio12", "bio15"),
  scenario = "ssp126",
  period   = "2011-2040",
  gcm      = "GFDL-ESM4",
  aoi      = aoi
)

# Compare historic vs future bio1 for Costa Rica
bio1_hist <- get_chelsa_historic(var = "bio1", aoi = aoi)
bio1_diff <- bio1_fut - bio1_hist
terra::plot(bio1_diff, main = "Temperature change (bio1): 2041-2070 SSP585")

# Extract per H3 hexagon
h7      <- paisaje::get_h3_grid(aoi, res = 7)
h7_fut  <- extract_num_raster(bio_stack, h7)

## End(Not run)
```

---

get_chelsa_historic *Download and Process Historic Climate Variables from CHELSA v2.1*

Description

Downloads historic bioclimatic variables from CHELSA v2.1 (Climatologies at High Resolution for the Earth's Land Surface Areas) for the reference period 1981-2010. The data are served as Cloud Optimized GeoTIFFs (COGs) from the Swiss WSL EnviCloud object store, which allows this function to retrieve only the spatial subset covering `aoi` without downloading the global file (~110 MB per variable).

One or more bioclimatic variables (bio1-bio19) can be requested in a single call. The result is a multi-layer `SpatRaster` optionally cropped and masked to the AOI, consistent with the interface of `get_worldclim_historic`.

Usage

```
get_chelsa_historic(
  var = "bio1",
  aoi = NULL,
  destination_dir = NULL,
  timeout = 300
)
```

Arguments

`var` ‘character‘ vector. One or more CHELSA bioclimatic variable names. Accepted values: `"bio1"` through `"bio19"`, or `"all"` (downloads all 19 variables). Variable names are case-insensitive. Default: `"bio1"`.

- `bio1` – Annual mean temperature (degC x 10)
- `bio2` – Mean diurnal temperature range
- `bio3` – Isothermality
- `bio4` – Temperature seasonality
- `bio5` – Max temperature of warmest month
- `bio6` – Min temperature of coldest month
- `bio7` – Temperature annual range
- `bio8` – Mean temperature of wettest quarter
- `bio9` – Mean temperature of driest quarter
- `bio10` – Mean temperature of warmest quarter
- `bio11` – Mean temperature of coldest quarter
- `bio12` – Annual precipitation (kg m-2 yr-1)
- `bio13` – Precipitation of wettest month
- `bio14` – Precipitation of driest month
- `bio15` – Precipitation seasonality
- `bio16` – Precipitation of wettest quarter
- `bio17` – Precipitation of driest quarter
- `bio18` – Precipitation of warmest quarter
- `bio19` – Precipitation of coldest quarter

`aoi` ‘sf‘ or ‘SpatVector‘ or ‘NULL‘. Area of interest used to crop and mask the raster. If `NULL` (default), the global raster is returned. Providing an AOI is strongly recommended – each CHELSA variable is ~110 MB globally, and the COG format enables efficient spatial subsetting.

`destination_dir` ‘character‘ or ‘NULL‘. Directory where the output `.tif` will be saved. If `NULL` (default), a temporary directory is used and a message is emitted.

`timeout` ‘numeric‘. Maximum time in seconds for each HTTP request. Default: `300`.

Details

Spatial resolution CHELSA v2.1 is provided at a **fixed resolution of 30 arc-seconds (~1 km)** globally. Unlike `get_worldclim_historic`, there is no `res` parameter – CHELSA does not offer coarser resolutions (2.5, 5, or 10 arc-minutes). If you need multi-resolution data, use `get_worldclim_historic` instead, or downsample the CHELSA output with `terra::aggregate()`.

Why CHELSA over WorldClim? CHELSA v2.1 and WorldClim v2.1 are both high-resolution (~1 km) global climatologies, but differ in their downscaling methodology:

- CHELSA uses a **mechanistic downscaling** approach based on atmospheric dynamics and orographic effects, which tends to perform better in complex terrain (mountains, coasts).

- WorldClim uses **statistical interpolation** (thin-plate splines), which is faster but less physically grounded.
- For tropical regions with complex topography (e.g., Costa Rica), CHELSA is generally considered more accurate.

COG streaming – no full download required CHELSA files are Cloud Optimized GeoTIFFs hosted on the Swiss WSL EnviCloud. This function uses the `/vsicurl/` virtual filesystem prefix from GDAL (via `terra`) to stream only the tiles that cover `aoi`, avoiding downloading the entire global file. When `aoi` is provided, spatial subsetting is done in memory before writing to disk.

Reference period All historic CHELSA v2.1 bioclimatic variables use the **1981-2010** climatological normal period. For future projections see `get_chelsa_future`.

Value

A `SpatRaster` with one layer per requested variable, named after the CHELSA filename convention (e.g., `CHELSA_bio1_1981-2010_V.2.1`). If `aoi` is provided, the raster is cropped and masked to it. Also written to `destination_dir` as a single multi-layer `.tif`. Returns `NULL` invisibly on error.

References

Karger, D. N., Conrad, O., Bohner, J., Kawohl, T., Kreft, H., Soria-Auza, R. W., Zimmermann, N. E., Linder, P., & Kessler, M. (2017). Climatologies at high resolution for the earth's land surface areas (CHELSA). *Scientific Data*, 4, 170122. doi:10.1038/sdata.2017.122

Brun, P., Zimmermann, N. E., Hari, C., Pellissier, L., & Karger, D. N. (2022). Global climate-related predictors at kilometre resolution for the past and future. *Earth System Science Data*, 14, 5573-5603. doi:10.5194/essd1455732022

See Also

- `get_chelsa_future` – CHELSA future projections (CMIP6).
- `get_worldclim_historic` – WorldClim v2.1 historic data.
- `extract_num_raster` – extract area-weighted means per polygon.
- CHELSA website: `https://chelsa-climate.org`
- EnviCloud browser: `https://envicloud.wsl.ch/#/?bucket=https://os.zhdk.cloud.switch.ch/chelsav2/`

Examples

```
## Not run:
library(sf)

# Use Costa Rica outline (included in paisaje)
aoi <- paisaje::cr_outline_c

# Single variable -- Annual mean temperature
bio1 <- get_chelsa_historic(var = "bio1", aoi = aoi)

# Multiple variables
```

```
bio_stack <- get_chelsa_historic(var = c("bio1", "bio12", "bio15"), aoi = aoi)

# All 19 bioclimatic variables
bio_all <- get_chelsa_historic(var = "all", aoi = aoi)

# Extract mean values per H3 hexagon
h7      <- paisaje::get_h3_grid(aoi, res = 7)
h7_clim <- extract_num_raster(bio_stack, h7)

## End(Not run)
```

`get_cr_outline` *Download Costa Rica Outline from GADM*

Description

Downloads the Costa Rica country boundary from GADM 4.1 via `gadm` and returns it as an `sf` object. By default returns only the continental landmass, excluding the Isla del Coco and other minor oceanic islands. The downloaded file is cached locally to avoid repeated downloads.

This function provides a reproducible, always up-to-date alternative to the static `cr_outline_c` dataset included in the package, and ensures consistency across packages that depend on a Costa Rica AOI (e.g., **paisaje** and **h3sdm**).

Usage

```
get_cr_outline(continental = TRUE, path = tempdir())
```

Arguments

`continental`	‘logical‘. If `TRUE` (default), returns only the continental landmass by selecting the polygon with the largest area. If `FALSE`, returns the full GADM boundary including all islands (Isla del Coco and minor Pacific/Caribbean islands).
`path`	‘character‘. Directory where the GADM file will be cached. Defaults to `tempdir()`. For persistent caching across sessions, provide a permanent directory (e.g., `"data/"`).

Details

Caching GADM files are cached by `geodata::gadm()` in `path`. If the file already exists, it is loaded from disk without downloading again. For persistent caching, set `path` to a permanent directory.

Consistency across packages Both **paisaje** and **h3sdm** use Costa Rica as the default study area in examples and vignettes. Using `get_cr_outline()` in both packages ensures the same geometry is used, derived from the same GADM version, rather than relying on static copies that may diverge over time.

Island exclusion When `continental = TRUE`, the Isla del Coco (~550 km offshore in the Pacific Ocean) and any other minor islands are excluded by retaining only the polygon with the largest area after casting to individual `POLYGON` geometries.

Value

An sf object with one feature (POLYGON geometry) in WGS 84 (EPSG:4326), representing the Costa Rica outline.

References

Global Administrative Areas (GADM) version 4.1. https://gadm.org/license.html

See Also

- cr_outline_c — static continental outline (no islands).
- cr_outline — static full outline (with islands).
- gadm — underlying download function.
- get_h3_grid — generate H3 grid over the returned AOI.
- GADM: https://gadm.org

Examples

```
## Not run:
# Continental outline only (default)
cr <- get_cr_outline()
plot(sf::st_geometry(cr), main = "Costa Rica (continental)")

# Full outline including islands
cr_full <- get_cr_outline(continental = FALSE)
plot(sf::st_geometry(cr_full), main = "Costa Rica (all territory)")

# Use as AOI - consistent with h3sdm
h7  <- get_h3_grid(cr, res = 7)
bio <- get_chelsa_historic(var = "bio1", aoi = cr)

# Persistent cache
cr <- get_cr_outline(path = "data/gadm")

## 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 speed.

Land-cover classification (ESA WorldCover 10 m v200):

| Value | Class (English) | Categoría (Español) | |:——:|:———————————|:————————————————| | 10 | Tree cover | Cobertura arbórea | | 20 | Shrubland | Matorrales | | 30 | Grassland | Pastizales / herbazales | | 40 | Cropland | Tierras de cultivo | | 50 | Built-up | Áreas construidas / urbanas | | 60 | Bare / Sparse vegetation | Vegetación escasa o suelos desnudos | | 70 | Snow and ice | Nieve y hielo permanentes | | 80 | Permanent water bodies | Cuerpos de agua permanentes | | 90 | Herbaceous wetland | Humedales herbáceos | | 95 | Mangroves | Manglares | | 100 | Moss and lichen | Musgos y líquenes |

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 from NASA Black Marble*

Description

Downloads NASA Black Marble nighttime lights raster data for a given area of interest (`aoi_sf`), time period, and temporal resolution (daily, monthly, or annual). Internally wraps `bm_raster` from the **blackmarbler** package (World Bank), which handles tile discovery, download, mosaicing, and cloud/quality masking automatically.

The result is a `SpatRaster` cropped and masked to the AOI, ready to be passed directly to `extract_num_raster` or any `terra`-based workflow in **paisaje**.

Usage

```
get_nightlight_data(
  aoi_sf,
  year,
  month = NULL,
  product_id = "VNP46A3",
  bearer,
  variable = NULL,
  quality_flag_rm = NULL,
  destination_dir = NULL,
  timeout = 1200
)
```

Arguments

`aoi_sf` ‘sf‘. An `sf` object defining the area of interest (AOI). Can be any polygon geometry (country outline, H3 grid extent, custom boundary). The raster will be cropped and masked to this extent.

`year` ‘numeric‘ or ‘character‘. The year of interest (e.g., `2022`). Used together with `month` to build the `date` argument passed to `bm_raster`.

`month` ‘numeric‘ or `NULL`. Month of the year (1–12). Required when `product_id = "VNP46A3"` (monthly composite). Ignored for annual products (`"VNP46A4"`). Default is `NULL`.

`product_id` ‘character‘. NASA Black Marble product identifier. Available options:

- `"VNP46A1"` — Daily, at-sensor radiance (500 m).
- `"VNP46A2"` — Daily, BRDF-corrected and gap-filled (500 m). Default variable: `Gap_Filled_DNB_BRDF-Corrected_NTL`.
- `"VNP46A3"` — **Monthly** composite, snow-free (500 m). Default variable: `NearNadir_Composite_Snow_Free`.
- `"VNP46A4"` — **Annual** composite, snow-free (500 m). Default variable: `NearNadir_Composite_Snow_Free`.

Default: `"VNP46A3"` (monthly).

bearer
: ‘character‘. NASA LAADS DAAC bearer token required for authentication. Obtain a free token at https://ladsweb.modaps.eosdis.nasa.gov/ under *Login > Generate Token*. It is strongly recommended to store this token as an environment variable (e.g., `NASA_BEARER`) and retrieve it with `Sys.getenv("NASA_BEARER")` rather than hardcoding it in scripts.

variable
: ‘character‘ or `NULL`. Specific variable (layer) to extract from the HDF5 product. If `NULL` (default), the package default for each `product_id` is used (see parameter description above). Pass `""` to trigger an informative error listing all valid variable names.

quality_flag_rm
: ‘integer vector‘ or `NULL`. Quality flag values for which pixels will be set to `NA`. Lower quality values can be removed to reduce noise. Default is `NULL` (no quality filtering).

destination_dir
: ‘character‘ or `NULL`. Directory where the output `.tif` and intermediate HDF5 tiles will be cached. If `NULL` (default), the system’s temporary directory (`tempdir`) is used and a message is emitted.

timeout
: ‘numeric‘. Maximum time in seconds allowed for HTTP downloads. Temporarily overrides `getOption("timeout")` and restores the original value on exit. Default: `1200` (20 minutes).

Details

Why NASA Black Marble over EOG? The previous implementation downloaded a global monthly raster (~500 MB) from the Earth Observation Group (EOG, Colorado School of Mines) via HTML scraping, regardless of the AOI size. NASA Black Marble improves on this in several ways:

- **AOI-aware**: only downloads the MODIS/VIIRS tiles that intersect `aoi_sf`, dramatically reducing download size for regional studies.
- **Higher scientific quality**: applies lunar irradiance modeling, atmospheric correction, BRDF correction, and cloud masking at the algorithm level (not post-hoc).
- **Daily, monthly, and annual** products under a unified interface.
- **Stable API**: accesses NASA LAADS DAAC via token-authenticated `httr2` requests — no fragile HTML scraping.
- **Resolution**: 500 m (vs. ~750 m for the EOG monthly_notile product).

Bearer token setup The NASA bearer token is free but required. Recommended setup:

```
# In .Renviron (open with usethis::edit_r_environ()):
NASA_BEARER=your_token_here

# Then in your script:
bearer <- Sys.getenv("NASA_BEARER")
```

Integration with paisaje The returned `SpatRaster` is ready to be passed directly to `extract_num_raster` to summarize nightlight values per polygon or H3 hexagon grid.

Value

A SpatRaster object cropped and masked to aoi_sf, containing the requested nighttime lights variable. Layer name reflects the product and date. Also written to destination_dir as a .tif. Returns NULL invisibly if an error occurs, with an informative message.

References

Román, M. O., et al. (2018). NASA's Black Marble nighttime lights product suite. *Remote Sensing of Environment*, 210, 113–143. doi:10.1016/j.rse.2018.03.017

Marty, R., & Vicente, G. S. (2024). *blackmarbler: Georeferenced Rasters and Statistics of Nighttime Lights from NASA Black Marble*. R package v0.2.5. World Bank. https://worldbank.github.io/blackmarbler/

See Also

- bm_raster — underlying download function.
- extract_num_raster — extract area-weighted means per polygon.
- get_worldclim_historic — analogous function for climate data.
- get_esa_10m — analogous function for land cover data.
- NASA Black Marble portal: https://blackmarble.gsfc.nasa.gov/
- LAADS DAAC token: https://ladsweb.modaps.eosdis.nasa.gov/

Examples

```
## Not run:
library(sf)

# Use Costa Rica outline (included in paisaje)
aoi <- paisaje::cr_outline_c

# Bearer token from environment variable (recommended)
bearer <- Sys.getenv("NASA_BEARER")

# Monthly composite - March 2022
ntl <- get_nightlight_data(
  aoi_sf     = aoi,
  year       = 2022,
  month      = 3,
  product_id = "VNP46A3",
  bearer     = bearer
)

# Annual composite - 2021
ntl_anual <- get_nightlight_data(
  aoi_sf     = aoi,
  year       = 2021,
  product_id = "VNP46A4",
  bearer     = bearer
)
```

```
# Extract mean nightlight per H3 hexagon
h7     <- paisaje::get_h3_grid(aoi, res = 7)
h7_ntl <- extract_num_raster(ntl, h7)

## End(Not run)
```

get_records — *Query Species Occurrence Records within an Area of Interest (AOI)*

Description

Downloads species occurrence records from providers (e.g., GBIF) using the spocc package, filtering the initial query by the exact polygonal boundary of the Area of Interest (AOI) for maximum efficiency and precision.

Usage

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

Arguments

species	Character string specifying the species name to query (e.g., "Puma concolor").
aoi_sf	An sf object defining the Area of Interest (AOI). Its CRS will be transformed to WGS84 (EPSG:4326) before query.
providers	Character vector of data providers to query (e.g., "gbif", "inat"). If NULL (default), all available providers are used.
limit	Numeric. The maximum number of records to retrieve per provider. Default is 500.
remove_duplicates	Logical. If TRUE, records with identical longitude and latitude are removed using dplyr::distinct(). Default is FALSE.
date	Character vector specifying a date range (e.g., c('2000-01-01', '2020-12-31')).

Details

The function transforms the aoi_sf polygon into a WKT string, which is used in the spocc::occ geometry argument. This method is more efficient than querying by the rectangular bounding box, as it reduces the number of irrelevant records downloaded. Final spatial filtering is performed using sf::st_intersection to ensure strict containment.

Value

An sf object of points containing the filtered occurrence records, with geometry confirmed to fall strictly within the aoi_sf boundary.

Examples

```
## Not run:
# Assuming aoi_sf is a valid sf polygon
# puma_records <- get_records("Puma concolor", aoi_sf, providers = "gbif", limit = 1000)
# head(puma_records)

## End(Not run)
```

---

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