Package ‘fastbioclim’

September 8, 2025

Type Package

Title Scalable and Efficient Derivation of Bioclimatic Variables

Version 0.2.1

Maintainer Gonzalo E. Pinilla-Buitrago <gepinillab@gmail.com>

Description Provides a high-performance framework for deriving bioclimatic and custom summary variables from large-scale climate raster data. The package features a dual-backend architecture that intelligently switches between fast in-memory processing for smaller datasets (via the 'terra' package) and a memory-safe tiled approach for massive datasets that do not fit in RAM (via 'exactextractr' and 'Rfast'). The main functions, 'derive_bioclim()' and 'derive_statistics()', offer a unified interface with advanced options for custom time periods and static indices, making it suitable for a wide range of ecological and environmental modeling applications. A software note is in preparation. In the meantime, you can visit the package website <https://gepinillab.github.io/fastbioclim/> to find tutorials in English and Spanish.

License GPL (>= 3)

URL https://gepinillab.github.io/fastbioclim/

BugReports https://github.com/gepinillab/fastbioclim/issues

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.1)

Imports exactextractr, future.apply, glue, progressr, purrr, qs, Rfast, rlang, sf, terra (>= 1.7-0)

NeedsCompilation no

Author Gonzalo E. Pinilla-Buitrago [aut, cre] (ORCID: <https://orcid.org/0000-0002-0065-945X>), Luis Osorio-Olvera [aut] (ORCID: <https://orcid.org/0000-0003-0701-5398>)

Repository CRAN

Date/Publication 2025-09-08 19:10:02 UTC

Contents

`bionames` *Print Bioclimatic Variable Names*

Description

This function prints the names of bioclimatic variables based on the specified indices.

Usage

```
bionames(bios = 1:35)
```

Arguments

`bios`	Numeric vector indicating the indices of bioclimatic variables to print. Default is 1:35, which prints all variable names.

Value

None. Prints the names of the selected bioclimatic variables to the console.

Examples

```
bionames()           # Print all bioclimatic variable names
bionames(c(1, 5, 12)) # Print names for variables 1, 5, and 12
```

`calculate_average` *Calculate Averages for SpatRasters*

Description

Calculates temporal averages for a multi-layer SpatRaster. This function serves as a smart wrapper, automatically selecting between an in-memory (‘terra‘) or out-of-core (‘tiled‘) workflow based on data size.

Usage

```
calculate_average(
  x,
  index,
  output_names = NULL,
  output_dir = tempdir(),
  user_region = NULL,
  method = c("auto", "tiled", "terra"),
  tile_degrees = 5,
  gdal_opt = c("COMPRESS=DEFLATE", "PREDICTOR=3", "NUM_THREADS=ALL_CPUS"),
  overwrite = FALSE
)
```

Arguments

`x`	A ‘terra::SpatRaster‘ object with multiple layers representing a time series.
`index`	A numeric or integer vector defining the grouping for aggregation. Its length must equal the number of layers in ‘x‘. For example, to average 360 monthly layers into 12 monthly means, ‘index‘ would be ‘rep(1:12, 30)‘.
`output_names`	A character vector of names for the output layers. Its length must equal the number of unique groups in ‘index‘. If ‘NULL‘, names like "avg_unit_1" are generated.
`output_dir`	The directory where the final averaged raster layers will be saved as GeoTIFF files.
`user_region`	(Optional) An ‘sf‘ or ‘terra::SpatVector‘ object for clipping.
`method`	The processing method: "auto", "tiled", or "terra".
`tile_degrees`	(Tiled method only) The approximate size of processing tiles.
`gdal_opt`	(Optional) GDAL creation options for the output GeoTIFFs.
`overwrite`	Logical. If ‘FALSE‘ (default), stops if output files exist.

Value

A ‘terra::SpatRaster‘ object pointing to the newly created files.

`calculate_roll` *Calculate Rolling Temporal Averages for SpatRasters*

Description

Calculates temporal summaries for each time unit over a moving window of cycles. This function is designed for time series where fundamental time **units** (e.g., months) are grouped into repeating **cycles** (e.g., years).

Usage

```
calculate_roll(
  x,
  window_size,
  freq = 12,
  step = 1,
  fun = "mean",
  name_template = "{prefix}_w{start_window}-{end_window}_u{idx_unit}",
  output_prefix = "output",
  output_dir = tempdir(),
  user_region = NULL,
  method = c("auto", "tiled", "terra"),
  tile_degrees = 5,
  gdal_opt = c("COMPRESS=DEFLATE", "PREDICTOR=3", "NUM_THREADS=ALL_CPUS"),
  overwrite = FALSE
)
```

Arguments

`x` A ‘terra::SpatRaster‘ object where each layer represents a time **unit**.

`window_size` Integer. The size of the moving window, measured in the number of **cycles**. For example, if the data cycle is annual (‘freq = 12‘), a ‘window_size‘ of 20 represents a 20-year window.

`freq` Integer. The number of time **units** (layers) that constitute one complete **cycle**. Common examples: 12 for monthly units in a yearly cycle, or 24 for hourly units in a daily cycle.

`step` Integer. The number of **cycles** to slide the window by for each iteration. Default is 1.

`fun` Character. The name of the summary function (e.g., "mean"). Default is "mean".

`name_template` A character string defining the template for output filenames, using ‘glue‘ syntax. Default: ‘"{prefix}_w{start_window}-{end_window}_u{idx_unit}"‘. Available placeholders are:

- ‘{prefix}‘: The value from ‘output_prefix‘.
- ‘{start_window}‘: The starting **cycle** index of the window.
- ‘{end_window}‘: The ending **cycle** index of the window.
- ‘{idx_unit}‘: The index of the time **unit** within the cycle (e.g., the month number).

`output_prefix` A character string for output filenames. Default is "output".

`output_dir` Directory to save the final GeoTIFF files.

`user_region` (Optional) An ‘sf‘ or ‘terra::SpatVector‘ for clipping.

`method` Processing method: "auto", "tiled", or "terra".

`tile_degrees` (Tiled method only) Approximate size of processing tiles.

`gdal_opt` (Optional) GDAL creation options for GeoTIFFs.

`overwrite` Logical. If ‘FALSE‘ (default), stops if output files exist.

Value

A ‘terra::SpatRaster‘ object pointing to the newly created files.

---

check_rasters *Validate Climate Input Rasters*

Description

Performs rigorous checks on a set of climate raster files to ensure they are suitable for bioclimatic variable calculation.

Usage

```
check_rasters(..., check_nas = TRUE)
```

Arguments

`...`	Named arguments providing character vectors of file paths for each climate variable (e.g., ‘tmin_files = c(...)‘, ‘prcp_files = c(...)‘).
`check_nas`	Logical. If ‘TRUE‘ (the default), perform the potentially slow check for mismatched NA values. If ‘FALSE‘, only check for geometry.

Details

This function checks for two main sources of error: 1. **Geometric Inconsistency**: Ensures all input rasters share the exact same coordinate reference system (CRS), extent, and resolution. 2. **NA Mismatch**: Checks if the pattern of NA values is consistent across all layers of all provided climate variables. Mismatched NAs can lead to silent errors in calculations. This check can be time-consuming for very large rasters.

Value

A list containing a summary of the validation:

`‘is_valid‘`	A single logical value: ‘TRUE‘ if all checks pass, ‘FALSE‘ otherwise.
`‘geom_report‘`	A message detailing the result of the geometry check.
`‘na_report‘`	A message detailing the result of the NA check.
`‘mismatch_raster‘`	If ‘check_nas=TRUE‘ and mismatches are found, a ‘SpatRaster‘ where non-zero values indicate pixels with inconsistent NAs.

`derive_bioclim` *Derive Comprehensive Bioclimatic Variables*

Description

Calculates up to 35 bioclimatic variables from average monthly climate SpatRasters (or other temporal units). This function serves as a smart wrapper that automatically selects the most efficient processing workflow (in-memory vs. tiled) based on data size and user-defined region of interest.

Usage

```
derive_bioclim(
  bios,
  tmin = NULL,
  tmax = NULL,
  prcp = NULL,
  tavg = NULL,
  srad = NULL,
  mois = NULL,
  output_dir = tempdir(),
  period_length = 3,
  circular = TRUE,
  user_region = NULL,
  method = c("auto", "tiled", "terra"),
  tile_degrees = 5,
  gdal_opt = c("COMPRESS=DEFLATE", "PREDICTOR=3", "NUM_THREADS=ALL_CPUS"),
  overwrite = FALSE,
  ...
)
```

Arguments

`bios` Numeric vector specifying which bioclimatic variables (1-35) to compute.

`tmin`, `tmax`, `prcp`, `tavg`, `srad`, `mois`
(Optional) ‘terra::SpatRaster‘ objects containing the climate data for each temporal unit (e.g., 12 monthly layers). All provided rasters must have the same geometry and number of layers.

`output_dir` The directory where the final bioclimatic variable rasters will be saved. The directory will be created if it does not exist. The default is temporal directory created by ‘tempdir‘.

`period_length` Integer. The number of temporal units (e.g., months) that define a "period" for calculating summary variables like BIO8 (Mean Temp of Wettest Quarter). Defaults to 3, representing quarters for monthly data.

`circular` Logical. If ‘TRUE‘ (the default), period calculations will wrap around the beginning and end of the time series (e.g., for monthly data, Dec-Jan-Feb is considered a valid period).

`user_region` (Optional) An ‘sf‘ or ‘terra::SpatVector‘ object defining the area of interest. If provided, all calculations will be clipped to this region.

`method` The processing method. See Details for more information.

`tile_degrees` (Tiled method only) The approximate size of processing tiles in degrees. Ignored if the ’terra’ workflow is used.

`gdal_opt` (Optional) A character vector of GDAL creation options for the output GeoTIFF files. Controls compression, threading, etc.

`overwrite` (Optional) Logical. If ‘FALSE‘ (the default), the function will stop immediately if any target output files already exist.

`...` Additional arguments, primarily for passing static index rasters. See the "Static Indices" section for details.

Details

This function unifies two processing backends. The ‘method‘ argument controls which is used:

- ‘"auto"‘: (Default) Intelligently chooses between "terra" and "tiled" based on estimated memory requirements.
- ‘"terra"‘: Forces the fast, in-memory workflow. May fail on very large datasets.
- ‘"tiled"‘: Forces the memory-safe, out-of-core workflow. Ideal for very large datasets. Requires that the input SpatRasters point to files on disk.

Period-based variables (e.g., BIO8, BIO10) are calculated using a moving window defined by ‘period_length‘.

Value

An SpatRaster with 35 bioclimatic variables or a subset of them:

bio01 Mean Temperature of Units

bio02 Mean Diurnal Range

bio03 Isothermality

bio04 Temperature Seasonality

bio05 Max Temperature of Warmest Unit

bio06 Min Temperature of Coldest Unit

bio07 Temperature Range of Units

bio08 Mean Temperature of Wettest Period

bio09 Mean Temperature of Driest Period

bio10 Mean Temperature of Warmest Period

bio11 Mean Temperature of Coldest Period

bio12 Precipitation Sum

bio13 Precipitation of Wettest Unit

bio14 Precipitation of Driest Unit

bio15 Precipitation Seasonality

bio16 Precipitation of Wettest Period

bio17 Precipitation of Driest Period

bio18 Precipitation of Warmest Period

bio19 Precipitation of Coldest Period

bio20 Mean Radiation of Units

bio21 Highest Radiation Unit

bio22 Lowest Radiation Unit

bio23 Radiation Seasonality

bio24 Radiation of Wettest Period

bio25 Radiation of Driest Period

bio26 Radiation of Warmest Period

bio27 Radiation of Coldest Period

bio28* Mean Moisture Content Of Units

bio29* Highest Moisture Content Unit

bio30* Lowest Moisture Content Unit

bio31* Moisture Content Seasonality

bio32* Mean Moisture Content of Most Moist Period

bio33* Mean Moisture Content of Least Moist Period

bio34* Mean Moisture Content of Warmest Period

bio35* Mean Moisture Content of Coldest Period

Static Indices

For advanced use cases, such as time-series analysis or defining specific seasons, you can provide pre-calculated index rasters to override the dynamic calculations. These are passed as named ‘SpatRaster‘ objects via the ‘...‘ argument (e.g., ‘warmest_period = my_warmest_idx_rast‘). The wrapper function automatically handles passing them to the appropriate workflow.

When using the "tiled" workflow, these static index rasters **must** be file-backed (i.e., not held entirely in memory). Supported static indices include:

- ‘warmest_unit‘, ‘coldest_unit‘, ‘wettest_unit‘, ‘driest_unit‘
- ‘high_rad_unit‘, ‘low_rad_unit‘, ‘high_mois_unit‘, ‘low_mois_unit‘
- ‘warmest_period‘, ‘coldest_period‘, ‘wettest_period‘, ‘driest_period‘
- ‘high_mois_period‘, ‘low_mois_period‘

Note

*The original moisture variables proposed in the ANUCLIM manual are based on the Moisture Index (MI). However, this function allows users to calculate moisture-based bioclimatic variables using other units of moisture as inputs, offering greater flexibility in input data usage.

References

O'Donnell, M. S., & Ignizio, D. A. (2012). Bioclimatic predictors for supporting ecological applications in the conterminous United States. ANUCLIM 6.1 User Guide. Centre for Resource and Environmental Studies, The Australian National University.

See Also

'validate_climate_inputs()' to check data integrity before processing.

Examples

```
# This is a conceptual example, requires data setup
## Not run:
  # Assume tmin_rast, tmax_rast, prcp_rast are 12-layer SpatRasters
  bioclim_vars <- derive_bioclim(
    bios = 1:19,
    tmin = tmin_rast,
    tmax = tmax_rast,
    prcp = prcp_rast,
    output_dir = "./bioclim_output",
    overwrite = TRUE
  )
  plot(bioclim_vars[[c("bio01", "bio12")]])

## End(Not run)
```

derive_statistics *Derive Custom Summary Statistics from Climate Variables*

Description

Calculates a wide range of custom summary statistics for a primary climate variable, with options for interactions with a second variable. This function serves as a smart wrapper that automatically selects the most efficient processing workflow (in-memory vs. tiled).

Usage

```
derive_statistics(
  variable,
  stats = c("mean", "max", "min"),
  inter_variable = NULL,
  inter_stats = NULL,
  prefix_variable = "var",
  suffix_inter_max = "inter_high",
  suffix_inter_min = "inter_low",
  output_dir = tempdir(),
  period_length = 3,
  period_stats = "mean",
  circular = TRUE,
  user_region = NULL,
  method = c("auto", "tiled", "terra"),
  tile_degrees = 5,
  gdal_opt = c("COMPRESS=DEFLATE", "PREDICTOR=3", "NUM_THREADS=ALL_CPUS"),
  overwrite = FALSE,
  ...
)
```

Arguments

variable A ‘terra::SpatRaster‘ object for the primary variable.

stats A character vector of statistics to compute for the primary variable. Supported: ‘"mean"‘, ‘"max"‘, ‘"min"‘, ‘"sum"‘, ‘"stdev"‘, ‘"cv_cli"‘, ‘"max_period"‘, ‘"min_period"‘.

inter_variable (Optional) A ‘terra::SpatRaster‘ for an interactive variable.

inter_stats (Optional) A character vector of interactive statistics to compute. Requires ‘inter_variable‘. Supported: ‘"max_inter"‘, ‘"min_inter"‘.

prefix_variable A character string used as the prefix for all output file names (e.g., ‘prefix_variable = "wind"‘ results in "wind_mean.tif", "wind_max.tif").

suffix_inter_max Character. Suffix for the "max_inter" statistic name. Default: "inter_high".

suffix_inter_min Character. Suffix for the "min_inter" statistic name. Default: "inter_low".

output_dir The directory where the final summary rasters will be saved.

period_length Integer. The number of temporal units defining a "period". Default: 3.

period_stats Character. The statistic ("mean" or "sum") to summarize data over each period. Default: "mean".

circular Logical. If ‘TRUE‘ (the default), period calculations wrap around.

user_region (Optional) An ‘sf‘ or ‘terra::SpatVector‘ object defining the area of interest.

method The processing method. See Details for more information.

tile_degrees (Tiled method only) The approximate size of processing tiles.

gdal_opt (Optional) A character vector of GDAL creation options for the output GeoTIFF files.

overwrite (Optional) Logical. If ‘FALSE‘ (the default), the function will stop if output files already exist.

... Additional arguments, primarily for passing static index ‘SpatRaster‘ objects. See the "Static Indices" section.

Details

This function provides a flexible alternative to ‘derive_bioclim()‘ for any multi-layer climate variable (e.g., wind speed, humidity). It unifies two processing backends, controlled by the ‘method‘ argument:

- ‘"auto"‘: (Default) Intelligently chooses between "terra" and "tiled".
- ‘"terra"‘: Forces the fast, in-memory workflow.
- ‘"tiled"‘: Forces the memory-safe, out-of-core workflow. Requires that all input SpatRasters point to files on disk.

Value

A ‘terra::SpatRaster‘ object pointing to the newly created summary rasters.

Static Indices

For advanced control, provide pre-calculated index rasters as named ‘SpatRaster‘ objects via the ‘...‘ argument (e.g., ‘max_unit = max_idx_rast‘). Supported indices: ‘max_unit‘, ‘min_unit‘, ‘max_period‘, ‘min_period‘, ‘max_interactive‘, ‘min_interactive‘.

Index