Package: ssmooth (via r-universe)

October 23, 2026

Title Smooth Raster Time Series

Version 0.1.0.9001

Description Smooth a sequence of 'terra' rasters using various algorithms (currently moving average, weighted moving average, and exponential smoothing). Also includes wrappers to smooth a vector time-series using these same algorithms. All smoothers use 'Rcpp' implementations for performance.

Imports terra, Rcpp

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 8.0.0

LinkingTo Rcpp

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://github.com/Biodiversity-Futures-Lab/ssmooth

BugReports https://github.com/Biodiversity-Futures-Lab/ssmooth/issues

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev libproj-dev libsqlite3-dev

Repository https://biodiversity-futures-lab.r-universe.dev

Date/Publication 2026-10-23 13:32:33 UTC

RemoteUrl https://github.com/biodiversity-futures-lab/ssmooth

RemoteRef HEAD

RemoteSha 25c020b5fe5e7615880b99acf445c86823ac2172

Contents

SmoothRasterTS 2
SmoothTS 3

Index **5**

SmoothRasterTS *Smooth Raster Time Series*

Description

Applies a smoothing operation to each pixel's time series in a multi-layer raster using one of three methods: mean (moving average), weighted moving average, or exponential moving average.

Usage

```
SmoothRasterTS(
  rast,
  ...,
  cores = 1L,
  filename = "",
  overwrite = FALSE,
  wopt = list()
)
```

Arguments

rast	SpatRaster. A multi-layer raster object where each layer represents a time point in the series.
...	Additional arguments passed to the SmoothTS function, such as method, n, weights, alpha, and n_init.
cores	Integer. Number of CPU cores to use for parallel processing. Default is 1. Don't change this unless you know what you're doing.
filename	Character. Optional filename for writing the output raster. If provided, the output raster will be written to this file.
overwrite	Logical. If TRUE and filename is provided, allows overwriting the existing file; optional.
wopt	List. Optional list of additional arguments to pass to terra::writeRaster.

Details

Assumes that the input raster has multiple layers, where each layer represents a time point in the series. The function applies the specified smoothing method to the time series of each pixel across the layers and returns a new raster with the smoothed time series for each pixel.

Value

SpatRaster. A multi-layer raster object with smoothed time series for each pixel.

Examples

```
rast <- terra::rast(nrow = 1, ncol = 2, nlyrs = 5)
terra::values(rast) <- matrix(rep(1:5, each = 2), nrow = 2)
out <- SmoothRasterTS(rast, method = "mean", n = 3)
```

`SmoothTS` *Smooth Time Series Data*

Description

Applies a smoothing operation to a numeric vector (time series data) using one of three methods: mean (moving average), weighted moving average, or exponential moving average.

Usage

```
SmoothTS(x, method = "exponential", ...)

SimpleMA(x, n = 3)

WeightedMA(x, weights = c(1, 1, 1))

ExponentialMA(x, alpha = 0.3, n_init = 5)
```

Arguments

`x`	Numeric vector. The time series data to be smoothed.
`method`	Character. Smoothing method to use. Options are `"mean"`, `"weighted"`, or `"exponential"`. Default is `"exponential"`.
`...`	Additional arguments passed to the specific smoothing method functions.
`n`	Integer. Number of points for the moving average (used in "mean" method). Default is 3.
`weights`	Numeric vector. Weights for the weighted moving average (used in "weighted" method). Default is a vector of 1's (i.e., simple moving average).
`alpha`	Numeric. Smoothing factor for exponential moving average (used in "exponential" method). Default is 0.3.
`n_init`	Integer. Number of initial points to use for the first smoothed value in the exponential moving average (used in "exponential" method). Default is 5.

Details

For the "mean" method (`SimpleMA`), a simple moving average is computed over a specified window size. For the "weighted" method (`WeightedMA`), a weighted moving average is computed using a specified set of weights. For the "exponential" method (`ExponentialMA`), an exponential moving average is computed using a specified smoothing factor (alpha) and a specified number of initial points to use for the first smoothed value.

If you use a windowing method (mean or weighted), the function pads the input vector with the mean of the non-NA values at the beginning and end to ensure that the output vector has the same length as the input.

We include functions to run these smoothers individually, if you so desire, without using the main `SmoothTS` function.

Value

Numeric vector of smoothed values.

Examples

```
x <- 1:10
SmoothTS(x, method = "mean", n = 3)
SmoothTS(x, method = "weighted", weights = c(0.1, 0.5, 0.4))
SmoothTS(x, method = "exponential", alpha = 0.5, n_init = 3)
```

Index

ExponentialMA *(*SmoothTS*)*, 3

SimpleMA *(*SmoothTS*)*, 3
SmoothRasterTS, 2
SmoothTS, 3

WeightedMA *(*SmoothTS*)*, 3