Package: sgolay (via r-universe)

June 10, 2024

Type Package

Title Efficient Savitzky-Golay Filtering

Version 1.0.3

Date 2023-03-30

URL https://github.com/zeehio/sgolay

BugReports https://github.com/zeehio/sgolay/issues

Description Smoothing signals and computing their derivatives is a common requirement in signal processing workflows. Savitzky-Golay filters are a established method able to do both (Savitzky and Golay, 1964 <doi:10.1021/ac60214a047>). This package implements one dimensional Savitzky-Golay filters that can be applied to vectors and matrices (either row-wise or column-wise). Vectorization and memory allocations have been profiled to reduce computational fingerprint. Short filter lengths are implemented in the direct space, while longer filters are implemented in frequency space, using a Fast Fourier Transform (FFT).

Imports signal

License GPL (>= 2)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Suggests covr, RUnit

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

RemoteUrl https://github.com/zeehio/sgolay

RemoteRef HEAD

RemoteSha be05d0956169bdbf790e6e1310da87c634fc77a1

Contents

sgolayfilt *Apply a Savitzky-Golay smoothing filter*

Description

Smooth data or compute its derivatives with a Savitzky-Golay smoothing filter.

Usage

```
sgolayfilt(
  x,
  p = 3,
  n = p + 3 - p%%2,
  m = 0,
  ts = 1,
  rowwise = FALSE,
  engine = c("auto", "fft", "filter")
)
```

Arguments

`x` A numeric matrix or vector

`p` filter order.

`n` filter length (must be odd).

`m` return the m-th derivative of the filter coefficients.

`ts` time scaling factor.

`rowwise` If `TRUE`, Apply the filter by rows instead of by columns

`engine` How is the filter applied. This parameter impacts the performance, but not the results. "auto" will select automatically an efficient engine. `"fft"` uses a Fast Fourier Transform to apply the filter. `"filter"` uses a convolution in the direct space. `"fft"` is more efficient on larger filter lengths.

Value

A matrix or vector of the same dimensions or length as `x`, with the result of the filter

Examples

```
x <- runif(300)
y <- sgolayfilt(x, p=2, n = 21)
```

Index