Package: svFast (via r-universe)

May 26, 2026

Type Package

Version 0.1.0

Title 'SciViews::R' - Fast and Parallelized R Functions

Description A Series of Fast Math and Stat Functions, Parallelized using 'RcppParallel'.

Maintainer Philippe Grosjean <phgrosjean@sciviews.org>

Depends R (>= 4.2.0), stats (>= 4.2.0)

Imports Rcpp (>= 1.0.10), RcppParallel (>= 5.1.7)

Suggests bench (>= 1.1.0), covr (>= 3.5.0), knitr (>= 1.42), rmarkdown (>= 2.21), spelling (>= 2.2.1), testthat (>= 3.0.0)

LinkingTo Rcpp, RcppParallel

SystemRequirements GNU make, Intel TBB, Windows: cmd.exe and cscript.exe, Solaris: g++ is required

License MIT + file LICENSE

URL https://github.com/SciViews/svFast, https://www.sciviews.org/svFast/, https://sciviews.r-universe.dev/svFast

BugReports https://github.com/SciViews/svFast/issues

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

VignetteBuilder knitr

Encoding UTF-8

Language en-US

ByteCompile yes

Config/testthat/edition 3

Config/pak/sysreqs make

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

Date/Publication 2025-08-29 06:17:22 UTC

RemoteUrl https://github.com/SciViews/svFast

RemoteRef HEAD

RemoteSha a3f5b888463d5afd4b93eae945b1bb8165ec6431

Contents

`Log_` *Logarithm (Fast Parallel Version)*

Description

Fast version of logarithmic and exponential functions (when vector size >= 50000). `log_()` computes the natural logarithm of `x` (base `e` by default), `log2_()` computes the base 2 logarithm, `log10_()` computes the base 10. `log1p_()` computes `log(1 + x)` accurately even for small `x`.

`exp_()` computes the exponential function. `expm1_()` computes `exp(x) - 1` accurately even for small `x`.

Usage

```
log10_(x, para = 50000L)

log2_(x, para = 50000L)

log1p_(x, para = 50000L)

exp_(x, para = 50000L)

expm1_(x, para = 50000L)

log_(x, base = 2.71828182845905, para = 50000L)
```

Arguments

`x`	vector of numeric values
`para`	the minimum length of `x` to use parallel computation (50000 by default)
`base`	the base of the logarithm (e = `exp(1)` by default)

Details

They are **not** generic functions and do not process factor, Date, POSIXt, difftime, complex, or S4 objects (use base R equivalent function instead). Data frames are processed column-wise, providing each column is compatible. All attributes are preserved.

Value

A numeric vector, matrix, or data frame with the transformed values.

See Also

`log10()`, `log2()`, `log()`, `log1p()`, `exp()`, `expm1()`

Examples

```
log_(1:5)
log_(1:5, base = 2.5)
```

Round_ *Rounding of Numbers (Fast Parallel Version)*

Description

Fast version of rounding (when vector size >= 50000). `ceiling_()` takes a single numeric argument x and returns a numeric vector containing the smallest integers not less than the corresponding elements of x. It is similar to `ceiling()`.

`floor_()` takes a single numeric argument x and returns a numeric vector containing the largest integers not greater than the corresponding elements of x. It is similar to `floor()`.

`trunc_()` takes a single numeric argument x and returns a numeric vector containing the integers formed by truncating the values in x toward 0. It is similar to `trunc()`.

`round_()` rounds the values in its first argument to the specified number of decimal places (default 0). See 'Details' about "round to even" in `round()` when rounding off a 5.

`signif_()` rounds the values in its first argument to the specified number of significant digits.

Usage

```
ceiling_(x, para = 50000L)

floor_(x, para = 50000L)

trunc_(x, para = 50000L)

round_(x, digits = 0L, para = 50000L)

signif_(x, digits = 6L, para = 50000L)
```

Arguments

x	vector of numeric values
para	the minimum length of x to use parallel computation (50000 by default)
digits	integer indication the number of decimal places (round_()) or significant digits (signif_()) to be used. For round_(), negative values are allowed and indicate rounding to a power of ten (-2 means rounding to the nearest hundred), see round().

Details

They are **not** generic functions and do not process factor, Date, POSIXt, difftime, complex, or S4 objects (use base R equivalent function instead). Data frames are processed column-wise, providing each column is compatible. All attributes are preserved.

Value

A numeric vector, matrix, or data frame with the transformed values.

See Also

ceiling(), floor(), trunc()

round(), signif()

Examples

```
floor_(c(1.23, 4.56, -7.89))
```

Trig_ *Trigonometric Functions (Fast Parallel Version)*

Description

Fast version of trigonometric functions (when vector size >= 50000). They respectively compute the cosine, sine, tangent, arc-cosine, arc-sine, arc-tangent, and the two-argument arc-tangent.

cospi_(x), sinpi_(x), and tanpi_(x), compute cos(pi*x), sin(pi*x), and tan(pi*x).

Usage

```
cos_(x, para = 50000L)

sin_(x, para = 50000L)

tan_(x, para = 50000L)

acos_(x, para = 50000L)
```

```
asin_(x, para = 50000L)

atan_(x, para = 50000L)

cospi_(x, para = 50000L)

sinpi_(x, para = 50000L)

tanpi_(x, para = 50000L)
```

Arguments

x	vector of numeric values
para	the minimum length of x to use parallel computation (50000 by default)

Details

They are **not** generic functions and do not process factor, Date, POSIXt, difftime, complex, or S4 objects (use base R equivalent function instead). Data frames are processed column-wise, providing each column is compatible. All attributes are preserved.

Value

A numeric vector, matrix, or data frame with the transformed values.

See Also

`cos()`, `sin()`, `tan()`, `acos()`, `asin()`, `atan()`, `atan2()`

Examples

```
cos_(1:5)
```

Index