

Package ‘detrendr’

February 18, 2018

Type Package

Title Detrend Images

Version 0.4.0

Maintainer Rory Nolan <rorynolan@gmail.com>

Description Image series affected by bleaching must be corrected by 'detrending' prior to the performance of quantitative analysis. 'detrendr' is for correctly detrending images. It uses Nolan's algorithm (Nolan et al., 2017 <doi:10.1093/bioinformatics/btx434>).

License BSD_3_clause + file LICENSE

Encoding UTF-8

LazyData true

LinkingTo Rcpp, RcppParallel

SystemRequirements GNU make

Imports Rcpp, RcppParallel, parallel, checkmate, magrittr, dplyr, foreach, doParallel, RSAGA, iterators, purrr, jrtiff (>= 0.3.0), autothresholdr (>= 1.1.1), rlang, stringr, filesstrings (>= 2.0.4)

URL <https://www.github.com/rorynolan/detrendr>

BugReports <https://www.github.com/rorynolan/detrendr/issues>

RoxygenNote 6.0.1

Suggests testthat, knitr, rmarkdown, covr, abind

VignetteBuilder knitr

NeedsCompilation yes

Author Rory Nolan [aut, cre, cph],
Sergi Padilla-Parra [ths, cph]

Repository CRAN

Date/Publication 2018-02-18 04:52:16 UTC

R topics documented:

apply_on_pillars	2
best_degree	3
best_l	4
best_tau	5
brightness_pillars	6
detrend-directory	6
detrended_img	8
detrending	9
detrendr	11
pillar-stats	12
Index	13

apply_on_pillars	<i>Apply a function to each pillar of a 3-dimensional array.</i>
------------------	--

Description

Define a 'pillar' of a 3-dimensional array as pillar i, j of array `arr` being `arr[i, j,]`. This function applies a specified function to each pillar.

Usage

```
apply_on_pillars(arr3d, FUN)
```

Arguments

<code>arr3d</code>	A 3-dimensional array.
<code>FUN</code>	A function which takes a vector as input and, for a given input length, outputs a vector of constant length (can be 1).

Value

If `FUN` is returning length 1 vectors, a matrix whereby `mat[i, j] = FUN(arr3d[i, j,])`.
 If `FUN` is returning vectors of length $l > 1$, a 3-dimensional array whereby `arr[i, j,] = FUN(arr3d[i, j,])`.

best_degree	<i>Find the best polynomial degree for polynomial detrending.</i>
-------------	---

Description

Use Nolan's algorithm to find the ideal polynomial degree for polynomial detrending.

Usage

```
best_degree(img, seed = NULL, parallel = FALSE)
```

Arguments

img	A 4-dimensional array in the style of an <code>ijtiff_img</code> (indexed by <code>img[y, x, channel, frame]</code>) or a 3-dimensional array which is a single channel of an <code>ijtiff_img</code> (indexed by <code>img[y, x, frame]</code>).
seed	Random numbers may be generated during the detrending process. For reproducibility, you can set a seed for this random number generation here.
parallel	Would you like to use multiple cores to speed up this function? If so, set the number of cores here, or to use all available cores, use <code>parallel = TRUE</code> .

Value

If no detrend is necessary, this function returns NA. If a detrend is required, this function returns a natural number which is the ideal polynomial degree for polynomial detrending. If there are multiple channels, the function returns a vector, one degree parameter for each channel.

References

Rory Nolan, Luis A. J. Alvarez, Jonathan Elegheert, Maro Iliopoulou, G. Maria Jakobsdottir, Marina Rodriguez-Muñoz, A. Radu Aricescu, Sergi Padilla-Parra; `nandb`—number and brightness in R with a novel automatic detrending algorithm, *Bioinformatics*, <https://doi.org/10.1093/bioinformatics/btx434>.

Examples

```
## Not run:  
## These examples are not run on CRAN because they take too long.  
## You should still try them for yourself.  
img <- ijrtiff::read_tif(system.file('extdata', 'bleached.tif',  
                                  package = 'detrndr'))  
best_degree(img, seed = 0, parallel = 2)  
  
## End(Not run)
```

`best_l`*Find the best length parameter for boxcar detrending.*

Description

Use Nolan's algorithm to find the ideal length parameter for boxcar detrending. Boxcar detrending is also referred to as 'running average'.

Usage

```
best_l(img, seed = NULL, parallel = FALSE)
```

Arguments

<code>img</code>	A 4-dimensional array in the style of an <code>ijtiff_img</code> (indexed by <code>img[y, x, channel, frame]</code>) or a 3-dimensional array which is a single channel of an <code>ijtiff_img</code> (indexed by <code>img[y, x, frame]</code>).
<code>seed</code>	Random numbers may be generated during the detrending process. For reproducibility, you can set a seed for this random number generation here.
<code>parallel</code>	Would you like to use multiple cores to speed up this function? If so, set the number of cores here, or to use all available cores, use <code>parallel = TRUE</code> .

Value

If no detrend is necessary, this function returns NA. If a detrend is required, this function returns a natural number which is the ideal length parameter for boxcar detrending. If there are multiple channels, the function returns a vector, one `l` parameter for each channel.

References

Rory Nolan, Luis A. J. Alvarez, Jonathan Elegheert, Maro Iliopoulou, G. Maria Jakobsdottir, Marina Rodriguez-Muñoz, A. Radu Aricescu, Sergi Padilla-Parra; nandb—number and brightness in R with a novel automatic detrending algorithm, *Bioinformatics*, <https://doi.org/10.1093/bioinformatics/btx434>.

Examples

```
## These examples are not run on CRAN because they take too long.
## You should still try them for yourself.
img <- ijtiff::read_tif(system.file('extdata', 'bleached.tif',
                                  package = 'detrendr'))
best_l(img, seed = 0, parallel = 2)
```

best_tau	<i>Find the best tau parameter for exponential smoothing detrending.</i>
----------	--

Description

Use Nolan's algorithm to find the ideal tau parameter for exponential smoothing detrending.

Usage

```
best_tau(img, cutoff = 0.05, seed = NULL, parallel = FALSE)
```

Arguments

img	A 4-dimensional array in the style of an ijtiff_img (indexed by <code>img[y, x, channel, frame]</code>) or a 3-dimensional array which is a single channel of an ijtiff_img (indexed by <code>img[y, x, frame]</code>).
cutoff	In <i>exponential filtering</i> detrending, for the weighted average, every point gets a weight. This can slow down the computation massively. However, many of the weights will be approximately zero. With cutoff, we say that any point with weight less than or equal to cutoff times the maximum weight may be ignored; so with <code>cutoff = 0.05</code> , any weight less than 5% of the maximum weight may be ignored. The default value of this parameter is sensible and its value should not be set to anything else without good reason.
seed	Random numbers may be generated during the detrending process. For reproducibility, you can set a seed for this random number generation here.
parallel	Would you like to use multiple cores to speed up this function? If so, set the number of cores here, or to use all available cores, use <code>parallel = TRUE</code> .

Value

If no detrend is necessary, this function returns NA. If a detrend is required, this function returns a natural number which is the ideal tau parameter for exponential smoothing detrending. If there are multiple channels, the function returns a vector, one tau parameter for each channel.

References

Rory Nolan, Luis A. J. Alvarez, Jonathan Elegheert, Maro Iliopoulou, G. Maria Jakobsdottir, Marina Rodriguez-Muñoz, A. Radu Aricescu, Sergi Padilla-Parra; nandb—number and brightness in R with a novel automatic detrending algorithm, *Bioinformatics*, <https://doi.org/10.1093/bioinformatics/btx434>.

Examples

```
img <- ijtiff::read_tif(system.file('extdata', 'bleached.tif',
                                   package = 'detrendr'))[, , 1, ]
best_tau(img, seed = 0, parallel = 2)
```

brightness_pillars *Get the brightness of pillars of a 3d array.*

Description

For an [ijtiff_img](#)-style array `img` (indexed as `img[y, x, channel, frame]`), 3-dimensional array `mat3d`, pillar `xy` of channel `ch` is defined as `img[y, x, ch,]`. This function computes the brightness, of each pillar.

Usage

```
brightness_pillars(img, parallel = FALSE)
```

Arguments

`img` A 4-dimensional array in the style of an [ijtiff_img](#) (indexed by `img[y, x, channel, frame]`) or a 3-dimensional array which is a single channel of an [ijtiff_img](#) (indexed by `img[y, x, frame]`).

`parallel` Would you like to use multiple cores to speed up this function? If so, set the number of cores here, or to use all available cores, use `parallel = TRUE`.

Value

An [ijtiff_img](#)-style array `arr` with one frame. `arr[y, x, ch, 1]` is equal to `var(img[y, x, ch,]) / mean(img[y, x, ch,])`

Examples

```
aaa <- array(1:16, dim = c(2, 2, 4))
brightness_pillars(aaa)
```

detrend-directory *Detrend all TIFF images in an entire folder.*

Description

Batch processing. Apply any of the available detrending routines to detrend all of the TIFF images in a folder, saving the detrended images as TIFF files in the same folder.

Usage

```

dir_detrend_boxcar(folder_path = ".", l, purpose = c("FCS", "FFS"),
  thresh = NULL, seed = NULL, parallel = FALSE, msg = TRUE)

dir_detrend_exp(folder_path = ".", tau, purpose = c("FCS", "FFS"),
  thresh = NULL, seed = NULL, parallel = FALSE, msg = TRUE)

dir_detrend_polynom(folder_path = ".", degree, purpose = c("FCS", "FFS"),
  thresh = NULL, seed = NULL, parallel = FALSE, msg = TRUE)

```

Arguments

folder_path	The path (relative or absolute) to the folder you wish to process.
l	The length parameter for <i>boxcar</i> detrending. The size of the sliding window will be $2 * l + 1$. This must be a positive integer. Set this to "auto" to use Nolan's algorithm to automatically find a suitable value for this parameter (recommended). For multi-channel images, it is possible to have a different l for each channel by specifying l as a vector or list.
purpose	What type of calculation do you intend to perform on the detrended image? If it is an FFS (fluorescence fluctuation spectroscopy) calculation (like number and brightness), choose 'FFS'. If it is an FCS (fluorescence correlation spectroscopy) calculation (like cross-correlated number and brightness or autocorrelation), choose 'FCS'. The difference is that if purpose is 'FFS', the time series is corrected for non-stationary mean and variance, whereas if purpose is 'FCS', the time series is corrected for non-stationary mean only.
thresh	The threshold or thresholding method (see autothresholdr::mean_stack_thresh()) to use on the image prior to detrending.
seed	Random numbers may be generated during the detrending process. For reproducibility, you can set a seed for this random number generation here.
parallel	Would you like to use multiple cores to speed up this function? If so, set the number of cores here, or to use all available cores, use parallel = TRUE.
msg	Receive messages to tell you how the processing of the directory is going? Default is yes.
tau	The <i>tau</i> parameter for <i>exponential filtering</i> detrending. This must be a positive number. Set this to "auto" to use Nolan's algorithm to automatically find a suitable value for this parameter (recommended). For multi-channel images, it is possible to have a different tau for each channel by specifying tau as a vector or list.
degree	The degree of the polynomial to use for the polynomial detrending. This must be a positive integer. Set this to "auto" to use Nolan's algorithm to automatically find a suitable value for this parameter (recommended). For multi-channel images, it is possible to have a different degree for each channel by specifying degree as a vector or list.

Details

These functions include a thresholding option, unlike their non-batch processing counterparts which they wrap (i.e. `img_detrend_boxcar`, `img_detrend_exp` and `img_detrend_polynom`). This is because, when working interactively, it's easy to threshold and then detrend, but for batch processing, it's not so easy to efficiently do one after the other, so it's nice to have that taken care of should you want it.

Value

Silently, a character vector of the paths to the detrended images.

Examples

```
## Not run:
setwd(tempdir())
file.copy(c(system.file("extdata", "bleached.tif", package = "detrendr"),
             system.file("img", "2ch_ij.tif", package = "ijtiff")),
         ".")
dir_detrend_boxcar(l = "auto", thresh = "tri", purpose = "FFS")
dir_detrend_exp(tau = "auto", thresh = "tri", purpose = "FCS")
dir_detrend_polynom(degree = "auto", thresh = "huang", purpose = "FFS")
## End(Not run)
```

detrended_img

Detrended image class.

Description

A `detrended_img` is a 4-dimensional array of positive integers in the style of an `ij TIFF_img` (indexed by `img[y, x, channel, frame]`) which is the result of a detrending routine. It has 4 attributes:

`method` The detrending method used. This must be one of "boxcar", "exponential" or "polynomial".

`parameter` The value of the parameter used. This will be the `l`, `tau` or `degree` parameter for the respective methods.

`auto` A boolean that is TRUE if the parameter was found automatically or FALSE if it was manually selected.

`purpose` Either "FCS" or "FFS" to denote whether the detrending was done for the purpose of fluorescence correlation spectroscopy or fluorescence fluctuation spectroscopy calculations respectively.

Usage

```
detrended_img(img, method, parameter, auto, purpose = c("FCS", "FFS"))
```


Arguments

img	The detrended image series. A 4-dimensional array of non-negative integers in the style of an ijtiff_img , or a 3-dimensional array of non-negative integers which represents a single channel of an ijtiff_img -style array (indexed by <code>img[y, x, frame]</code>).
method	The method used. One of "boxcar", "exponential" or "polynomial".
parameter	A number. The detrend parameter used. One per channel.
auto	Logical. Was automatic detrending used? One per channel.
purpose	Was the image detrended for the purpose of doing FCS or FFS calculations? See detrending .

Details

Sometimes when detrending, you can get slight negative values in the detrended image. These values should really just be zero, so this constructor function sets negative values of `img` to zero.

Value

An object of class `detrended_img`.

detrending	<i>Detrend images.</i>
------------	------------------------

Description

Correct images for bleaching (or any other effect that introduces an unwanted trend) by *detrending*.

Usage

```
img_detrend_boxcar(img, l, purpose = c("FCS", "FFS"), seed = NULL,
  parallel = FALSE)
```

```
img_detrend_exp(img, tau, cutoff = 0.05, purpose = c("FCS", "FFS"),
  seed = NULL, parallel = FALSE)
```

```
img_detrend_polynom(img, degree, purpose = c("FCS", "FFS"), seed = NULL,
  parallel = FALSE)
```

Arguments

img	A 4-dimensional array in the style of an ijtiff_img (indexed by <code>img[y, x, channel, frame]</code>) or a 3-dimensional array which is a single channel of an ijtiff_img (indexed by <code>img[y, x, frame]</code>).
-----	---

l	The length parameter for <i>boxcar</i> detrending. The size of the sliding window will be $2 * l + 1$. This must be a positive integer. Set this to "auto" to use Nolan's algorithm to automatically find a suitable value for this parameter (recommended). For multi-channel images, it is possible to have a different l for each channel by specifying l as a vector or list.
purpose	What type of calculation do you intend to perform on the detrended image? If it is an FFS (fluorescence fluctuation spectroscopy) calculation (like number and brightness), choose 'FFS'. If it is an FCS (fluorescence correlation spectroscopy) calculation (like cross-correlated number and brightness or autocorrelation), choose 'FCS'. The difference is that if purpose is 'FFS', the time series is corrected for non-stationary mean and variance, whereas if purpose is 'FCS', the time series is corrected for non-stationary mean only.
seed	Random numbers may be generated during the detrending process. For reproducibility, you can set a seed for this random number generation here.
parallel	Would you like to use multiple cores to speed up this function? If so, set the number of cores here, or to use all available cores, use parallel = TRUE.
tau	The <i>tau</i> parameter for <i>exponential filtering</i> detrending. This must be a positive number. Set this to "auto" to use Nolan's algorithm to automatically find a suitable value for this parameter (recommended). For multi-channel images, it is possible to have a different tau for each channel by specifying tau as a vector or list.
cutoff	In <i>exponential filtering</i> detrending, for the weighted average, every point gets a weight. This can slow down the computation massively. However, many of the weights will be approximately zero. With cutoff, we say that any point with weight less than or equal to cutoff times the maximum weight may be ignored; so with cutoff = 0.05, any weight less than 5% of the maximum weight may be ignored. The default value of this parameter is sensible and its value should not be set to anything else without good reason.
degree	The degree of the polynomial to use for the polynomial detrending. This must be a positive integer. Set this to "auto" to use Nolan's algorithm to automatically find a suitable value for this parameter (recommended). For multi-channel images, it is possible to have a different degree for each channel by specifying degree as a vector or list.

Details

There are 3 detrending methods available: *boxcar*, *exponential filtering* and *polynomial*. These are described in detail in Nolan et al., 2017.

- *Boxcar* detrending with parameter l is a moving average detrending method using a sliding window of size $2l + 1$.
- *Exponential filtering* detrending is a moving weighted average method where for parameter τ the weights are calculated as $\exp(-t/\tau)$ where t is the distance from the point of interest.
- *Polynomial* detrending works by fitting a polynomial line to a series of points and then correcting the series to remove the trend detailed by this polynomial fit.

Value

The detrended image, an object of class `detrended_img`.

References

Rory Nolan, Luis A. J. Alvarez, Jonathan Elegheert, Maro Iliopoulou, G. Maria Jakobsdottir, Marina Rodriguez-Muñoz, A. Radu Aricescu, Sergi Padilla-Parra; nandb—number and brightness in R with a novel automatic detrending algorithm, *Bioinformatics*, <https://doi.org/10.1093/bioinformatics/btx434>.

Examples

```
## Not run:
## These examples are not run on CRAN because they take too long.
## You should still try them for yourself.
img <- ijttiff::read_tif(system.file('extdata', 'bleached.tif',
                                   package = 'detrendr'))
corrected <- img_detrend_boxcar(img, "auto", purpose = "fcs",
                               seed = 0, parallel = 2)
corrected10 <- img_detrend_boxcar(img, 10, purpose = "fcs",
                                  seed = 0, parallel = 2)
corrected50 <- img_detrend_boxcar(img, 50, purpose = "fcs",
                                  seed = 0, parallel = 2)
corrected <- img_detrend_exp(img, "auto", purpose = "ffs",
                             seed = 0, parallel = 2)
corrected10 <- img_detrend_exp(img, 10, purpose = "ffs",
                               seed = 0, parallel = 2)
corrected50 <- img_detrend_exp(img, 50, purpose = "fcs",
                               seed = 0, parallel = 2)
corrected <- img_detrend_polynom(img, "auto", purpose = "ffs",
                                 seed = 0, parallel = 2)
corrected2 <- img_detrend_polynom(img, 2, purpose = "ffs",
                                  seed = 0, parallel = 2)
## End(Not run)
```

detrendr

detrendr: Image detrending in R.

Description

The `detrendr` package gives functions for detrending images, most often used for preprocessing in fluorescence fluctuation and correlation spectroscopy (FFS and FCS).

References

Rory Nolan, Luis A. J. Alvarez, Jonathan Elegheert, Maro Iliopoulou, G. Maria Jakobsdottir, Marina Rodriguez-Muñoz, A. Radu Aricescu, Sergi Padilla-Parra; nandb—number and brightness in R with a novel automatic detrending algorithm, *Bioinformatics*, <https://doi.org/10.1093/bioinformatics/btx434>.

pillar-stats

Get the means/medians/variances of pillars of an [ijtiff_img](#)-style array.

Description

For an [ijtiff_img](#)-style array `img` (indexed as `img[y, x, channel, frame]`), pillar `xy` of channel `ch` is defined as `img[y, x, ch,]`. These functions compute the mean, median and variance of each pillar for each channel.

Usage

```
mean_pillars(img, parallel = FALSE)
```

```
median_pillars(img, parallel = FALSE)
```

```
var_pillars(img, parallel = FALSE)
```

Arguments

<code>img</code>	A 4-dimensional array in the style of an ijtiff_img (indexed by <code>img[y, x, channel, frame]</code>) or a 3-dimensional array which is a single channel of an ijtiff_img (indexed by <code>img[y, x, frame]</code>).
<code>parallel</code>	Would you like to use multiple cores to speed up this function? If so, set the number of cores here, or to use all available cores, use <code>parallel = TRUE</code> .

Value

An [ijtiff_img](#)-style array `arr` with one frame. `arr[y, x, ch, 1]` is equal to `mean(img[y, x, ch,])`, `median(img[y, x, ch,])`, or `var(img[y, x, ch,])`.

Examples

```
aaa <- array(seq_len(2 ^ 4), dim = rep(2, 4)) # a 2-channel, 2-frame array
mean_pillars(aaa)
median_pillars(aaa)
var_pillars(aaa)
```

Index

`apply_on_pillars`, 2
`autothresholdr::mean_stack_thresh()`, 7

`best_degree`, 3
`best_l`, 4
`best_tau`, 5
`brightness_pillars`, 6

`detrend-directory`, 6
`detrended_img`, 8, 8, 11
`detrending`, 9, 9
`detrendr`, 11
`detrendr-package (detrendr)`, 11
`dir_detrend_boxcar (detrend-directory)`, 6
`dir_detrend_exp (detrend-directory)`, 6
`dir_detrend_polynom (detrend-directory)`, 6

`ijtiff_img`, 3–6, 8, 9, 12
`img_detrend_boxcar`, 8
`img_detrend_boxcar (detrending)`, 9
`img_detrend_exp`, 8
`img_detrend_exp (detrending)`, 9
`img_detrend_polynom`, 8
`img_detrend_polynom (detrending)`, 9

`mean_pillars (pillar-stats)`, 12
`median_pillars (pillar-stats)`, 12

`pillar-stats`, 12

`var_pillars (pillar-stats)`, 12