

Package ‘LFApp’

July 28, 2021

Version 1.3

Date 2021-07-27

Title Shiny Apps for Lateral Flow Assays

Author Filip Paskali [aut, cre],

Weronika Schary [aut] (<<https://orcid.org/0000-0002-7229-316X>>),

Matthias Kohl [aut] (<<https://orcid.org/0000-0001-9514-8910>>)

Maintainer Filip Paskali <F.Paskali@hs-furtwangen.de>

Description Shiny apps for the quantitative analysis of images from lateral flow assays (LFAs). The images are segmented and background corrected and color intensities are extracted. The apps can be used to import and export intensity data and to calibrate LFAs by means of linear, loess, or gam models. The calibration models can further be saved and applied to intensity data from new images for determining concentrations.

License LGPL-3

Depends R (>= 4.0.0)

Imports stats, utils, graphics, methods, mgcv, shiny, shinyjs, shinythemes, shinyFiles, shinyMobile (>= 0.9), EBImage, DT, ggplot2, fs

Suggests knitr, rmarkdown, remotes

VignetteBuilder knitr

Encoding UTF-8

URL <https://github.com/fpaskali/LFApp>

BugReports <https://github.com/fpaskali/LFApp/issues>

NeedsCompilation no

Repository CRAN

Date/Publication 2021-07-28 11:00:05 UTC

R topics documented:

LFApp-package	2
run_functions	2
threshold_li	3
triangle	4

Index**6**

LFApp-package	<i>Shiny Apps for Lateral Flow Assays.</i>
---------------	--

Description

Shiny apps for the quantitative analysis of images from lateral flow assays (LFAs). The images are segmented and background corrected and color intensities are extracted. The apps can be used to import and export intensity data and to calibrate LFAs by means of linear, loess, or gam models. The calibration models can further be saved and applied to intensity data from new images for determining concentrations.

Details

Package: LFApp
 Type: Package
 Version: 1.3
 Date: 2021-07-27
 Depends: R(>= 4.0.0)
 Imports: stats, utils, graphics, methods, mgcv, shiny, shinyjs, shinythemes, shinyFiles, shinyMobile (>= 0.9), EBImage,
 Suggests: knitr, rmarkdown, remotes
 License: LGPL-3
 URL: <https://github.com/fpaskali/LFApp>
 BugReports: <https://github.com/fpaskali/LFApp/issues>

```
library(LFApp)
```

Author(s)

Filip Paskali, Weronika Schary, Matthias Kohl
 Maintainer: Filip Paskali <F.Paskali@hs-furtwangen.de>

run_functions	<i>Run Analysis Shiny Apps</i>
---------------	--------------------------------

Description

Function start the Analysis Shiny App.

Usage

```
run_analysis()  
run_cal()  
run_core()  
run_quan()  
run_mobile_analysis()  
run_mobile_cal()  
run_mobile_core()  
run_mobile_quan()
```

Details

The functions start the various shiny apps included in the package.

Value

An object that represents the app. Printing the object will run the app.

Author(s)

Filip Paskali <F.Paskali@hs-furtwangen.de>, Weronika Schary <W.Schary@hs-furtwangen.de>, Matthias Kohl <Matthias.Kohl@stamats.de>

Examples

```
if(interactive()){  
  ## start full analysis app  
  run_analysis()  
  ## start mobile version of full analysis app  
  run_mobile_analysis()  
}
```

threshold_li

Li Thresholding Algorithm

Description

The function computes a background threshold of an image by using Li's iterative minimum cross entropy method.

Usage

```
threshold_li(image, tolerance = NULL, initial_guess = NULL, iter_callback = NULL)
```

Arguments

image object of class Image from package EBImage.
 tolerance optional tolerance threshold.
 initial_guess optional initial value for the minimization.
 iter_callback optional function applied to the minimization criterion.

Details

For more details about the method see Li and Lee (1993) as well as Li and Tam (1998).

Value

numeric vector with the computed threshold.

Author(s)

Filip Paskali <F.Paskali@hs-furtwangen.de>

References

C.H. Li and C.K. Lee (1993). Minimum cross entropy thresholding. *Pattern Recognition* **26** (4): 617-25. [https://doi.org/10.1016/0031-3203\(93\)90115-D](https://doi.org/10.1016/0031-3203(93)90115-D).

C.H. Li and P.K.S. Tam (1998). An iterative algorithm for minimum cross entropy thresholding. *Pattern Recognition Letters* **19** (8): 771-76. [https://doi.org/10.1016/S0167-8655\(98\)00057-9](https://doi.org/10.1016/S0167-8655(98)00057-9).

Examples

```
library(EBImage)
x <- readImage(system.file("images", "sample.TIF", package="LFApp"))
threshold_li(x)
```

triangle

Triangle Thresholding Algorithm

Description

The function computes a background threshold of an image using the triangle algorithm.

Usage

```
triangle(image, offset = 0.2, breaks = 256)
```

Arguments

image object of class Image from package EBImage.
 offset numeric, additional offset added to the computed threshold.
 breaks integer, number of breaks used in the histogram.

Details

The Triangle method is based on the histogram of the intensities. Based on the range of intensities and the maximum peak a threshold is determined. The method was proposed in Zack et al. (1977).

Value

numeric vector with the computed threshold.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

References

G.W. Zack, W. E. Rogers, and S. A. Latt (1977). Automatic measurement of sister chromatid exchange frequency. *The journal of histochemistry and cytochemistry: official journal of the Histochemistry Society* **25** (7): 741-53. <https://doi.org/10.1177/25.7.70454>.

Examples

```
library(EBImage)
x <- readImage(system.file("images", "sample.TIF", package="LFApp"))
triangle(x)
```

Index

- * **dynamic**

- run_functions, 2

- * **package**

- LFApp-package, 2

- * **univar**

- threshold_li, 3

- triangle, 4

LFApp (LFApp-package), 2

LFApp-package, 2

run_analysis (run_functions), 2

run_cal (run_functions), 2

run_core (run_functions), 2

run_functions, 2

run_mobile_analysis (run_functions), 2

run_mobile_cal (run_functions), 2

run_mobile_core (run_functions), 2

run_mobile_quan (run_functions), 2

run_quan (run_functions), 2

threshold_li, 3

triangle, 4