

Package ‘autothresholdr’

January 31, 2017

Type Package

Title An R port of the ImageJ plugin ``Auto Threshold''

Version 0.1.0

Maintainer Rory Nolan <rorynolan@gmail.com>

Description This package usrs rJava to implement ImageJ's thresholding methods in R.

License GPL-3

LazyData TRUE

Imports rJava,
RSAGA,
magrittr,
EBImage

SystemRequirements Java

RoxygenNote 5.0.1

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

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

R topics documented:

auto_thresh	1
Index	5

auto_thresh	<i>Automatically threshold an image.</i>
-------------	--

Description

These functions apply the ImageJ "Auto Threshold" plugin's image thresholding methods. The available methods are "IJDefault", "Huang", "Intermodes", "IsoData", "Li", "MaxEntropy", "Mean", "MinErrorI", "Minimum", "Moments", "Otsu", "Percentile", "RenyiEntropy", "Shanbhag", "Triangle", "Yen". Read about them at http://imagej.net/Auto_Threshold.

Usage

```
auto_thresh(int_arr, method, ignore.black = FALSE, ignore.white = FALSE)
```

```
auto_thresh_mask(int_arr, method, ignore.black = FALSE,
  ignore.white = FALSE)
```

```
auto_thresh_apply_mask(int_arr, method, fail = NA, ignore.black = FALSE,
  ignore.white = FALSE)
```

Arguments

<code>int_arr</code>	An array (or vector) of <i>integers</i> .
<code>method</code>	The name of the method you wish to use (e.g. "Huang"). Partial matching is performed i.e. <code>method = "h"</code> is enough to get you "Huang" and <code>method = "in"</code> is enough to get you "Intermodes".
<code>ignore.black</code>	Ignore black pixels/elements (zeros) when performing the thresholding?
<code>ignore.white</code>	Ignore white pixels when performing the thresholding? If set to <code>TRUE</code> , the function makes a good guess as to what the white (saturated) value would be (see "Details"). If this is set to a number, all pixels with value greater than or equal to that number are ignored.
<code>fail</code>	When using <code>auto_thresh_apply_mask</code> , to what value do you wish to set the pixels which fail to exceed the threshold.

Details

- NA values are automatically ignored.
- For `ignore.white = TRUE`, if the maximum value in the array is one of 2^8-1 , $2^{12}-1$, $2^{16}-1$ or $2^{32}-1$, then those max values are ignored. That's because they're the white values in 8, 12, 16 and 32-bit images respectively (and these are the common image bit sizes to work with). This guesswork has to be done because R does not know how many bits the image was on disk. This guess is very unlikely to be wrong, and if it is, the consequences are negligible anyway. If you're very concerned, then just specify the max value in the `ignore.white` argument.
- If you have set `ignore.black = TRUE` and/or `ignore.white = TRUE` but you are still getting error/warning messages telling you to try them, then your chosen method is not working for the given array, so you should try a different method.

Value

`auto_thresh` returns an integer, the image threshold value. Pixels exceeding this threshold are passed, but pixels at or below this level are failed.

`auto_thresh_mask` returns a binarized version of the input, with a value of `TRUE` at points which exceed the threshold and `FALSE` at those which do not. This has an attribute "threshold" to tell you what the threshold value was.

`auto_thresh_apply_mask` returns the original input masked by the threshold, i.e. all points not exceeding the threshold are set to a user-defined value (default NA). This has an attribute "threshold" to tell you what the threshold value was.

Acknowledgements

Gabriel Landini coded all of these functions together in one Java file for the ImageJ plugin.

References

- Huang, L-K & Wang, M-J J (1995), "Image thresholding by minimizing the measure of fuzziness", *Pattern Recognition* 28(1): 41-51
- Prewitt, JMS & Mendelsohn, ML (1966), "The analysis of cell images", *Annals of the New York Academy of Sciences* 128: 1035-1053
- Ridler, TW & Calvard, S (1978), "Picture thresholding using an iterative selection method", *IEEE Transactions on Systems, Man and Cybernetics* 8: 630-632
- Li, CH & Lee, CK (1993), "Minimum Cross Entropy Thresholding", *Pattern Recognition* 26(4): 617-625
- Li, CH & Tam, PKS (1998), "An Iterative Algorithm for Minimum Cross Entropy Thresholding", *Pattern Recognition Letters* 18(8): 771-776
- Sezgin, M & Sankur, B (2004), "Survey over Image Thresholding Techniques and Quantitative Performance Evaluation", *Journal of Electronic Imaging* 13(1): 146-165
- Kapur, JN; Sahoo, PK & Wong, ACK (1985), "A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram", *Graphical Models and Image Processing* 29(3): 273-285
- Glasbey, CA (1993), "An analysis of histogram-based thresholding algorithms", *CVGIP: Graphical Models and Image Processing* 55: 532-537
- Kittler, J & Illingworth, J (1986), "Minimum error thresholding", *Pattern Recognition* 19: 41-47
- Prewitt, JMS & Mendelsohn, ML (1966), "The analysis of cell images", *Annals of the New York Academy of Sciences* 128: 1035-1053
- Tsai, W (1985), "Moment-preserving thresholding: a new approach", *Computer Vision, Graphics, and Image Processing* 29: 377-393
- Otsu, N (1979), "A threshold selection method from gray-level histograms", *IEEE Trans. Sys., Man., Cyber.* 9: 62-66, doi:10.1109/TSMC.1979.4310076
- Doyle, W (1962), "Operation useful for similarity-invariant pattern recognition", *Journal of the Association for Computing Machinery* 9: 259-267, doi:10.1145/321119.321123
- Kapur, JN; Sahoo, PK & Wong, ACK (1985), "A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram", *Graphical Models and Image Processing* 29(3): 273-285
- Shanbhag, Abhijit G. (1994), "Utilization of information measure as a means of image thresholding", *Graph. Models Image Process.* (Academic Press, Inc.) 56 (5): 414-419, ISSN 1049-9652
- Zack GW, Rogers WE, Latt SA (1977), "Automatic measurement of sister chromatid exchange frequency", *J. Histochem. Cytochem.* 25 (7): 74153, PMID 70454
- Yen JC, Chang FJ, Chang S (1995), "A New Criterion for Automatic Multilevel Thresholding", *IEEE Trans. on Image Processing* 4 (3): 370-378, ISSN 1057-7149, doi:10.1109/83.366472
- Sezgin, M & Sankur, B (2004), "Survey over Image Thresholding Techniques and Quantitative Performance Evaluation", *Journal of Electronic Imaging* 13(1): 146-165

Examples

```
img <- EBImage::readImage(system.file("extdata", "eg.tif",
package = "autothresholdr"), as.is = TRUE)
auto_thresh(img, "h")
auto_thresh(img, "tri")
```

```
auto_thresh(img, "Otsu")
mask <- auto_thresh_mask(img, "h")
EBImage::display(mask, method = "r")
masked <- auto_thresh_apply_mask(img, "h")
EBImage::display(EBImage::normalize(masked), method = "r")
```

Index

`auto_thresh`, [1](#)
`auto_thresh_apply_mask (auto_thresh)`, [1](#)
`auto_thresh_mask (auto_thresh)`, [1](#)