NISTIR 7871

A New Measure in Cell Image Segmentation Data Analysis

Jin Chu Wu
Michael Halter
Raghu N. Kacker
John T. Elliott

http://dx.doi.org/10.6028/NIST.IR.7871

National Institute of Standards and Technology
U.S. Department of Commerce

AUTHENTICATED
U.S. GOVERNMENT
INFORMATION
GPO

NISTIR 7871

A New Measure in Cell Image Segmentation Data Analysis

Jin Chu Wu
Information Access Division, Information Technology Laboratory

Michael Halter
Biochemical Science Division, Material Measurement Laboratory

Raghu N. Kacker
Applied and Computational Mathematics Division, Information Technology Laboratory

John T. Elliott
Biochemical Science Division, Material Measurement Laboratory

http://dx.doi.org/10.6028/NIST.IR.7871

July 2012



U.S. Department of Commerce
Rebecca Blank, Acting Secretary

National Institute of Standards and Technology
Patrick D. Gallagher, Under Secretary of Commerce for Standards and Technology and Director

A New Measure in Cell Image Segmentation Data Analysis

Jin Chu Wu[a], Michael Halter[c], Raghu N. Kacker[b] and John T. Elliott[c]
[a]Information Access Division, [b]Applied and Computational Mathematics Division,
Information Technology Laboratory,
[c]Biochemical Science Division, Material Measurement Laboratory,
National Institute of Standards and Technology, Gaithersburg, MD 20899

Abstract

Cell image segmentation (CIS) is critical for quantitative imaging in cytometric analyses. The data derived after segmentation can be used to infer cellular function. To evaluate CIS algorithms, first for dealing with comparisons of single cells treated as two-dimensional objects, a misclassification error rate (MER) is defined as a weighted sum of the false negative rate and the false positive rate. Then, all cells' MERs are aggregated to constitute a new measure called the total error rate, which statistically takes account of the sizes of the cells in such a way that the weight on the result for an algorithm is higher if larger cells are not segmented correctly. This total error rate is used to measure the performance level of CIS algorithms. It was tested by applying ten CIS algorithms taken from the image processing toolkit ImageJ to 106 cells in our database. Furthermore, these cells with different sizes were manually segmented to be treated as the ground-truth cells. The test results were supported by the primitive pairwise comparison between two algorithms' MERs on all cells.

Keywords: Cell image segmentation; Measure; Total error rate; Total probability; Misclassification error rate; False negative rate; False positive rate.

1 Introduction

Cell image segmentation (CIS) analysis is critical for quantitative imaging in cytometric analyses. The data derived after segmentation could ultimately be used to infer cellular function, such as cell movement and cell behavior, which reveals cells' response to various conditions and external factors, and thus plays a critical role in molecular biology and cellular biochemistry.

Under different normal and pathological conditions, certain types of cells may migrate to entirely different parts of the organism. Hence, the investigation of cell movement and behavior is directly related to the research in areas such as oncology of tumor cell metastasis and invasion, cell embryology of neural crest cells migrating from the neural tube to various areas of the embryo and transforming into different structures, and so on [1].

Usually, algorithms are designed and developed to segment cells in fluorescent microscopy images. It is obvious that the performance of a segmentation algorithm can affect the quantitative results derived from an image. Thus, assessing the performance quality of an algorithm is very important. To do so, the images are typically segmented manually first. This operation results in the identification of pixels that belong to the cell and pixels that belong to the background. The cells are treated as two-dimensional objects. Thereafter, the algorithm is validated by comparing the output segmentation of the algorithm to the manual segmentation.

A cell in an image, regardless of whether it is segmented manually or using an algorithm, is identified by pixels belonging to the cell. Cells in a fluorescent microscopy image segmented manually by experts are treated as the ground-truth (GT) cells, whereas cells in an image segmented using an algorithm are named as the algorithm-detected (AD) cells. It is clear that the determination of GT cells is pivotal in evaluating CIS algorithms. In this article, the process of manual segmentation is based on the protocol as described in Appendix 1.

Generally speaking, the geometric relationship between a GT cell and the corresponding AD cell consists of three regions: 1) some part of the GT cell is also identified by the algorithm, named as the intersection region; 2) some part of the GT cell is missed by the algorithm, called as the false negative (FN) region; 3) some part of the AD cell is mistakenly picked up which does not belong to the GT cell, named as the false positive (FP) region.

In this article, if an algorithm detects a cell, that is manually segmented as one GT cell object, as several cells, then all these AD cells are counted as one AD cell; if an algorithm segments several cells, that are manually identified as different GT cells, as just one cell, then all these GT cells are treated as one GT cell. The issues regarding FN and FP regions in the CIS are similar to those in other applications, such as biometrics, speaker recognition evaluations, etc. [2, 3].

Different algorithms may have different criteria and methods to determine the boundary of a cell in a fluorescent microscopy image, and thus have different abilities to identify cells with respect to different cell features. For cells with some specific characteristics, some algorithms may perform better than others. As a result, how to measure the performance level of a CIS algorithm is a very important issue.

There are several metrics that can be applied to evaluate the performances of CIS methods[1]. They are, for instance, the Jaccard index [4], the Rand index [5], the Kappa statistic [6, 7], and so on [8]. However, each metric has its own advantages and disadvantages. For instance, the Jaccard and Rand indices do not take account of the spatial characteristics of segmentation [8]. The Kappa statistic could be an overly conservative measure of agreement [9].

In this article, to evaluate CIS algorithms, a new approach is proposed. The analysis of CIS starts with comparisons of single cells treated as two-dimensional objects using the misclassification error rate (MER) defined as a weighted sum of the FN rate and the FP rate. Then, all cells' MERs are aggregated to constitute a new measure called the total error rate, which statistically takes account of the sizes of the cells in such a way that the weight on the result for an algorithm is higher if larger cells are not segmented correctly. This total error rate is used to measure the performance level of CIS algorithms.

There are many factors that can affect how accurately a CIS algorithm detects the boundary of a cell in a fluorescent image. The cell size is one major factor. In general, large cells should be easier to detect than small cells, so the MER should be smaller for segmenting larger cells. If an algorithm is unable to detect larger cells well, it can affect the overall performance of the algorithm more negatively.

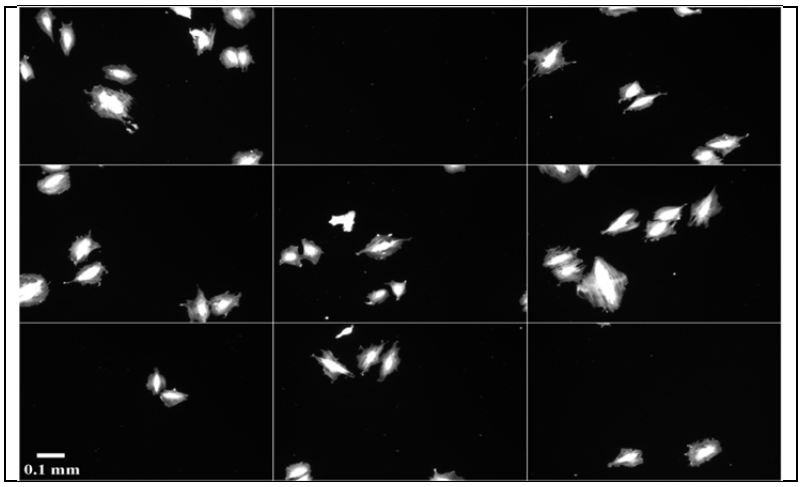


Figure 1 A Fluorescent image of some sample cells selected from 106 manually segmented cells.

The total error rate was tested on a dataset that consisted of 106 cells with different sizes, which were manually segmented to be taken as GT cells. These cells can be found in the NIST (National Institute of Standards and Technology) Semantics for Biological Data Resource: Cell Image Database [10]. In Figure 1, nine representative fluorescent microscopy images illustrate the data used in this article. As indicated in this figure, the sizes of cells vary.

[1] Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

The ten algorithms taken from the public domain and open source java-based image analysis package, ImageJ, were employed to segment cells in this article [11]. They are IJ_Huang, IJ_RenyiEntropy, IJ_Li, IJ_MaxEntropy, IJ_Intermodes, IJ_Minimum, IJ_Triangle, IJ_Yen, IJ_MinError, and IJ_Percentile, denoted by Algorithm 1 through 10 consecutively in the following text. Here the algorithms were numbered according to their performance levels in descending order (See Section 5). An ImageJ macro code for computing GT areas, AD areas, FPs and FNs in fluorescent microscopy images is provided in Appendix 2.

The MERs in the CIS data analysis are investigated in Section 2. Limitations of other approaches for evaluating CIS algorithms are discussed in Section 3. Our total error rate is defined in Section 4. The test results are presented in Section 5. Finally, the conclusions and discussion can be found in Section 6.

2 The MERs in the CIS data analysis

The first issue in the CIS data analysis in our approach is to define the MER for identifying a cell object in a fluorescent image using an automated algorithm. The sizes of a GT cell object and a related AD cell object are denoted by n_G and n_A, respectively. The sizes of the FN region, the FP region, and the intersection region as described in Section 1 are expressed by n_g, n_a, and n_I, respectively. All sizes are computed in terms of the number of pixels involved as discussed in Section 1.

These five parameters satisfy the following equations,

$$\begin{aligned} n_G &= n_I + n_g \\ n_A &= n_I + n_a \,. \end{aligned} \tag{1}$$

The FN rate r_{fn} and the FP rate r_{fp} are

$$\begin{aligned} r_{fn} &= \frac{n_g}{n_G} \\ r_{fp} &= \frac{n_a}{n_A}. \end{aligned} \tag{2}$$

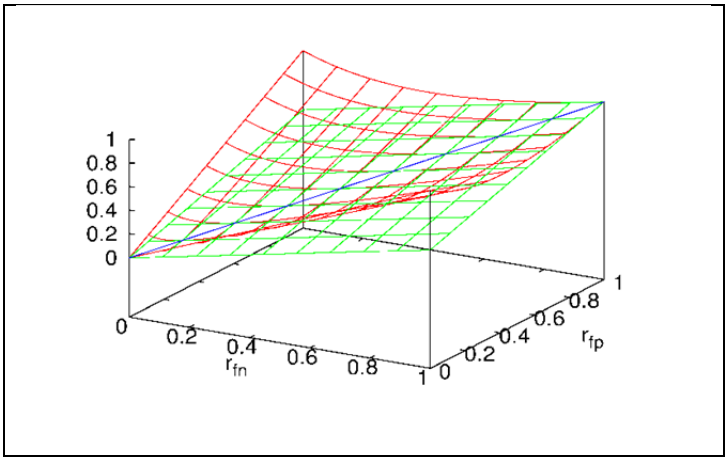


Figure 2 The weighted MER r_w is a surface (red) and the average MER r_a is a plane (green) with respect to the FN rate r_{fn} and the FP rate r_{fp}. They are tangent along a straight line (blue).

The MER of an algorithm with respect to detecting a cell is the proportion of pixels misclassified by the algorithm [12]. Therefore, several MERs can be defined in terms of the FN rate r_{fn} and the FP rate r_{fp}, by assigning different weights. Here are two of them:

$$r_a = \frac{r_{fn} + r_{fp}}{2}$$
$$r_w = \frac{r_{fn}^2 + r_{fp}^2}{r_{fn} + r_{fp}}. \quad (3)$$

The r_a is called the average MER with equal weight 1/2, and the r_w is named as the weighted MER using r_{fn} and r_{fp} themselves as weight so that the larger rate pays more penalties. As r_{fn} and r_{fp} approach to zero, r_w goes to zero as well. Both r_a and r_w vary in the region [0, 1]: 0 stands for the best segmentation when an AD cell is identical to the related GT cell, and 1 means the worst classification when an AD cell and the corresponding GT cell are disjoint.

However, some differences exist between r_w and r_a. First, the weighted MER r_w is a more conservative measure than the average MER r_a. It is trivial to prove using Eq. (3) that $r_w = r_a$ if and only if $r_{fn} = r_{fp}$; otherwise, $r_w > r_a$. This can also be seen in Fig. 2, where r_w is a surface in red and r_a is a plane in green as functions of the two variables r_{fn} and r_{fp}. The red surface is above the green plane except they are tangent along a straight line in blue.

Second, if an algorithm segments a small GT cell completely with a relatively very large AD cell, then $r_{fn} = 0$ and $r_{fp} \to 1$ and thus the weighted MER r_w approaches 1 but the average MER r_a goes to 1/2 due to Eq. (3). And also if an algorithm detects a large GT cell with a relatively very small AD cell located completely inside the GT cell, then $r_{fp} = 0$ and $r_{fn} \to 1$ and thus also $r_w \to 1$ but $r_a \to 1/2$. Indeed, when an AD cell contains a GT cell or is inside a GT cell and the size difference between the two cells is very large, the MER should be much larger than 1/2 and close to 1. The weighted MER r_w can deal with these special cases better than the average MER r_a.

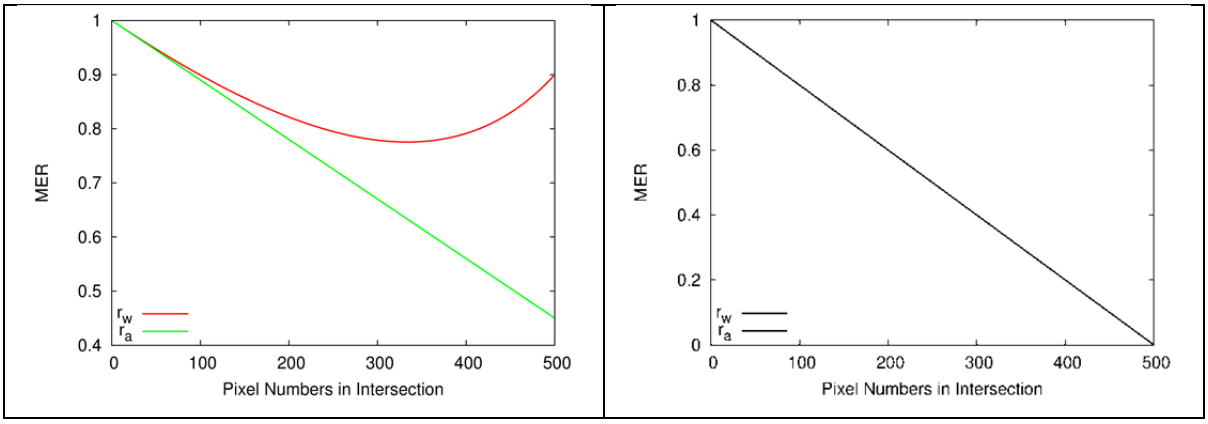


Figure 3 The weighted MER r_w and the average MER r_a as a function of the size of the intersection area n_I. (L): An AD cell with size 5000 (or 500) approaches to the related GT cell with size 500 (or 5000). (R): Both cells are in the size of 500.

Third, r_w is more rational than r_a, when the segmentation of an AD cell object is simulated to approach a GT cell object as the size of the intersection area n_I increases. To explore this feature, the average MER r_a and the weighted MER r_w are expressed as functions of n_I by

$$r_a = 1 - \frac{1}{2}\left(\frac{1}{n_G} + \frac{1}{n_A}\right) \times n_I \tag{4}$$

$$r_w = \frac{2 - 2\left(\frac{1}{n_G} + \frac{1}{n_A}\right) \times n_I + \left(\frac{1}{n_G^2} + \frac{1}{n_A^2}\right) \times n_I^2}{2 - \left(\frac{1}{n_G} + \frac{1}{n_A}\right) \times n_I}. \tag{5}$$

Since the GT cell size n_G and the AD cell size n_A are symmetric in Eqs. (4) and (5), r_a and r_w will vary in the same way regardless of whether $n_G > n_A$ or $n_G < n_A$.

In Fig. 3 (L) are depicted Eqs. (4) and (5) in which n_A and n_G were set to be 5000 and 500, respectively. As the intersection area n_I gets larger and larger, the average MER r_a decreases linearly all the way; but the weighted MER r_w decreases first and then increases. If an algorithm performs well, it should detect a cell with a comparable-size cell. It seems that the weighted MER r_w behaves more reasonably than the average MER r_a. The same argument holds true if n_A and n_G were set to be 500 and 5000, respectively, due to symmetry. When n_A is equal to n_G, the two MERs r_a and r_w are acting in the same way as shown in Fig. 3 (R) where both sizes are assumed to be 500.

In conclusion, the weighted MER r_w is a better measure than the average MER r_a for evaluating the performance of the CIS algorithms. However, in the following text, both of them are employed to illustrate the differences when used to compare algorithm segmentation with manual segmentation.

3 Limitations of other approaches for evaluating CIS algorithms

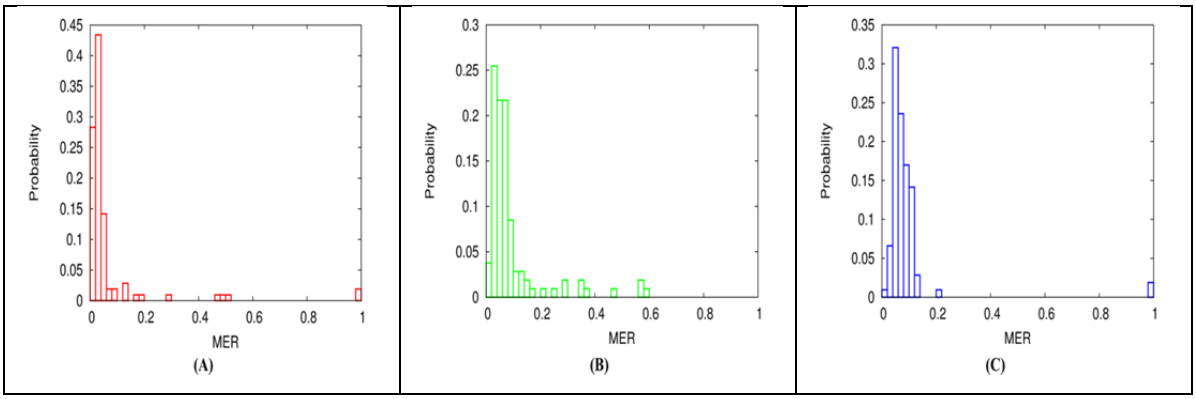


Figure 4 Histograms of the weighted MERs, r_w, generated using Algorithms 1 (A), 2 (B), and 3 (C).

The top three algorithms, Algorithms 1, 2, and 3 as identified in Section 1, were applied to segment 106 cells (see Section 5). Fig. 4 shows their MER histograms, in which the weighted MER r_w is used. These three histograms overlap each other. However, the histograms shift towards larger MER from Algorithm 1 to 3, suggesting that Algorithm 1 may be better than Algorithm 2 that in turn may be better than Algorithm 3.

When only the FN rate and FP rate are used to measure the performance level of a CIS algorithm, this is a bivariate issue. The three scatter plots of these two rates for Algorithms 1, 2, and 3 are shown in Fig. 5 (A). To enlarge the scatterplot, only the part from 0 to 0.3 for both FN rate and FP rate is shown. These three scatter plots overlap each other. Thus, it is impossible to distinguish which algorithm is better than the other. However, Algorithm 2 (in green) produces smaller FN rates but larger FP rates; and Algorithm 3 (in blue) is the other way around.

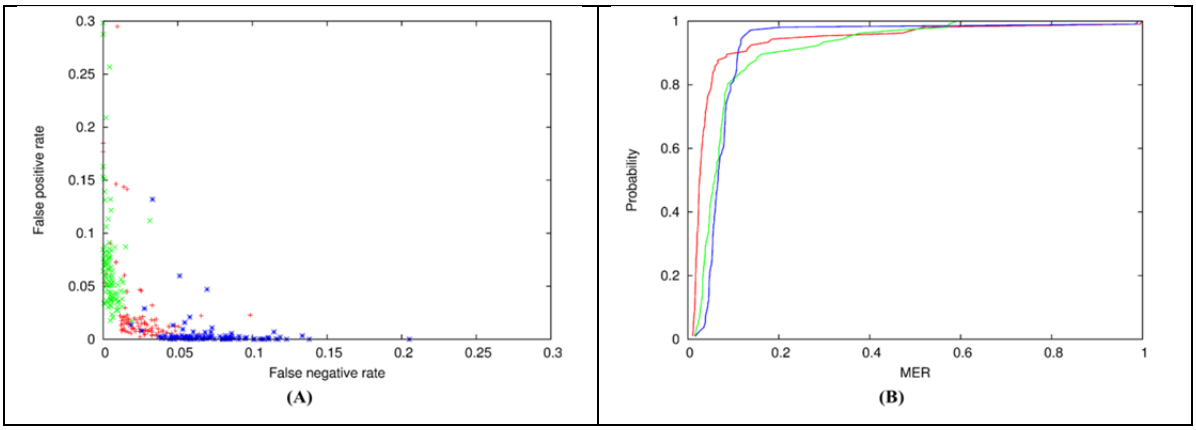


Figure 5 The scatter plots (A) and the CDFs (B) of the three algorithms. Algorithms 1, 2, and 3 are depicted in red, green, and blue, respectively. The Weighted MER is used.

If the cumulative distribution function (CDF) of the weighted MER r_w is employed, the three CDFs of Algorithms 1, 2, and 3 are depicted in Fig. 5 (B). These CDF curves cross repeatedly, making it impossible to quantitatively compare the performances of two CIS algorithms based on this approach either.

4 The total error rate in the CIS data analysis

To measure the performance level of a CIS algorithm, the statistic of interest is the total error rate ε, based on the formation of the total probability in statistics [13]. Generally speaking, segmenting a cell in fluorescent microscopy images is an exclusive event with respect to detecting other cells. Thus, the total error rate ε is defined as

$$\begin{aligned} \varepsilon \equiv \Pr(CIS) &= \sum_{i=1}^{n} \Pr(CIS \mid C_i) \Pr(C_i) \\ &= \sum_{i=1}^{n} MER_i \times \frac{S_i}{\sum_{j=1}^{n} S_j}. \end{aligned} \tag{6}$$

where n is the total number of cells, Pr (CIS) stands for the total probability of making misclassification errors while using an algorithm to detect all cells in a fluorescent image, the conditional probability Pr (CIS | C_i) means the MER while segmenting the i-th cell in the image which is denoted by MER_i, and Pr (C_i) is the probability of the occurrence of the i-th cell that is assumed to be the ratio of the size of the i-th GT cell S_i to the total sizes of all GT cell objects.

In Eq. (6), the MER_i can be either the weighted MER r_w or the average MER r_a as defined in Eq. (3) for segmenting the i-th cell in the image. It can be proven that the total error rate ε varies in the region [0, 1], where 0 stands for the best performance of the algorithm in the CIS and 1 means the worst performance.

Indeed, the total error rate ε aggregates all cell objects' MERs statistically to be a weighted sum using the sizes of the cell objects as weights. Such a formation of a measure in the CIS data analysis can ensure that the penalties for misclassifying cells are proportional to the sizes of cells.

5 Results

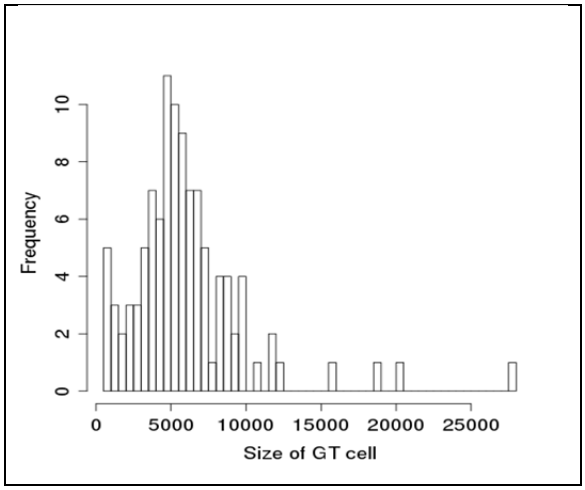


Figure 6 The histogram of the sizes of all 106 GT cells.

Algorithms		Weighted MER r_w is used the number of			Average MER r_a is used the number of		
		<	>	=	<	>	=
1	2	87	19	0	91	15	0
2	3	57	49	0	55	51	0
3	4	68	38	0	73	33	0
4	5	59	47	0	59	47	0
5	6	101	5	0	100	6	0
6	7	79	27	0	79	27	0
7	8	103	3	0	103	3	0
8	9	98	3	5	98	3	5
9	10	61	6	39	61	6	39

Table 1 Comparisons of two algorithms in terms of the numbers of inequalities and equalities for 106 cells using the weighted MER r_w and the average MER r_a, respectively.

The dataset employed in this article consisted of 106 cells with different sizes, which were manually segmented as GT cell objects [10]. Fig. 6 shows the histogram of sizes of all these 106 GT cell objects in terms of the total number of pixels covered by the GT cell object. The sizes of GT cell objects ranged from 647 up to 27 562. The median size was 5 368, and the mean value was 6 062. The distribution of the sizes of GT cells was skewed on the right side. The variation of sizes of the GT cells was quite large. Therefore, the sizes of the cells must be taken into account while designing the measure in the evaluation of CIS algorithms.

The MER, either the weighted MER r_w or the average MER r_a, can be computed for each of 106 cell objects using Eq. (3). The pairwise comparison can be conducted between the MERs of 106 cell objects generated using a CIS algorithm and those created using another algorithm. Table 1 shows the relationship in terms of the numbers of "less than" (<), "greater than" (>), and "equal to" (=) between the two corresponding MERs using two algorithms, where both r_w and r_a were employed.

The ten CIS algorithms taken from the ImageJ as defined in Section 1 were employed to generate MER for each cell object. In Table 1, Algorithm 1 was compared with Algorithm 2, Algorithm 2 was compared with Algorithm 3, and so on. For instance, comparing Algorithms 1 with 2, if the weighted MER r_w is used, there are 87 cells for which the MERs of Algorithm 1 are less than the MERs of Algorithm 2 and there are only 19 cells for which the situation is "greater than".

While comparing an algorithm against other algorithms having larger ordinal-number labels, the number of "less than" is still larger than the number of "greater than". For instance, while comparing Algorithm 1 against Algorithm 3, the numbers of "less than" and "greater than" are 91 and 15, respectively; while comparing Algorithm 1 against Algorithm 4, the numbers are 101 and 5, respectively, and so on. Generally speaking, when the "less than" occurs, the difference between the values of the two MERs gets larger and larger.

From this primitive inequality test, it suggests that the performance of Algorithm 1 be better than the performance of Algorithm 2, the performance of Algorithm 2 be better than the performance of Algorithm 3, and so on.

Algorithm	Total error rate ε using weighted MER	Total error rate ε using average MER
1	0.0575	0.0358
2	0.0669	0.0373
3	0.0894	0.0465
4	0.1051	0.0580
5	0.1712	0.0862
6	0.1735	0.0871
7	0.2244	0.1277
8	0.3750	0.2395
9	0.7133	0.5839
10	0.9742	0.9195

Table 2 The total error rates ε of CIS employing Algorithms 1 through 10 taken from the ImageJ, where both the weighted MER r_w and the average MER r_a were used.

The total error rates ε were computed using Eq. (6) for all ten CIS algorithms taken from the ImageJ, where both weighted MER r_w and average MER r_a were employed, as shown in Table 2. For instance, the total error rates ε of Algorithms 1 and 2 are 0.0575 and 0.0669, respectively, when r_w is used. It indicates that the segmentation masks of Algorithm 1 are more similar to the GT masks than Algorithm 2. The same argument can be applied to other cases as presented in Table 2.

It is clear that the conclusion based on the total error rate ε is fully supported by these primitive inequality tests. Moreover, regarding Algorithms 1, 2, and 3, the results shown in Table 2 are qualitatively consistent with the discussion in Section 2, where the histograms of MERs for these three algorithms shift gradually towards larger MER. Algorithms 9 and 10 generated a large fraction of MER equal to 1 (for both r_w and r_a), and thus their performances were very poor with total error rates higher than 0.5.

It is worth pointing out that in Table 2 for each algorithm the total error rate ε using the weighted MER r_w is larger than the one using the average MER r_a. This is because r_w is greater than or equal to r_a for the same FN rate and FP rate as discussed in Section 2, and the weights in terms of the sizes of GT cells are the same for both ways while calculating the total error rate as shown in Eq. (6). Again, the total error rate ε using r_w is more conservative than the one using r_a.

6 Conclusions and discussion

Evaluation of CIS algorithms starts with a single cell comparison using the MER defined in terms of the FN rate and the FP rate. The two types of MERs, i.e., the weighted MER r_w and the average MER r_a, were explored. The former is more conservative than the latter. It is more important that the former can deal with some special circumstances more suitably than the latter. Nonetheless, the test results using both of them were presented.

It is impossible to quantitatively evaluate CIS algorithms based on a bivariate criterion in terms of the FN rate and the FP rate by examining their scatter plots if the plots overlap. It is also difficult to do so by invoking the CDF curves of the MERs if the curves cross repeatedly. A method must be chosen to combine the computed FN rate and FP rate into a univariate measure for comparison.

In our case, the total error rate ε aggregates all cell objects' MERs based on the formation of the total probability in statistics. Indeed, it is a weighted sum of all cell objects' MERs by taking account of the sizes of the cells. In this way, the weight on the total error rate for an algorithm is higher if larger cells are not segmented correctly.

The total error rate ε was tested by applying the ten CIS algorithms taken from the ImageJ to our 106 cells with different sizes, which were manually segmented to be taken as the GT cell objects. The results were supported by the primitive pairwise comparison between the MERs of all these 106 cells generated using a CIS algorithm and those created using another algorithm, and also qualitatively consistent with the observations from their histograms.

As pointed out in Section 1, there are many factors that can affect how to segment cells in a fluorescent microscopy image and how to evaluate the performance of CIS algorithms. Our approach is based on detecting the boundary of cells treated as two-dimensional objects using cell sizes in terms of pixel numbers. Our method provides a way to measure the performance level of CIS algorithms under such circumstances. Further, certainly all numbers shown in Section 5 could be changed, if the protocol for generating manual segmentation treated as GT were refined. Nonetheless, this would have no substantial impact on the conclusions obtained in this article.

The sampling variability can result in measurement uncertainties. Thus, it is important to compute the uncertainty of the total error rate ε in terms of the standard error and the 95 % confidence interval. Subsequently, in order to see whether the difference between the performances of the two CIS algorithms is statistically significant, it is pertinent to conduct hypothesis testing. The research work on all these issues is underway.

Appendix 1: Protocol for generating manual segmentation [10]

1. Open ImageJ
2. Once in ImageJ, press “cntrl o” and select the frame to be manipulated. One can also press: File, then Open…
3. Click on “image” then roll over “type” then click on “RGB Color”.
4. Double click on color picker to open the color pallet.
5. Select either black (0,0,0) or yellow (255,255,0) or any other color as long as the same color is used to outline each cell.
6. Click on “image” then click on “adjust” and click on “Brightness and Contrast”.
7. Adjust the brightness of the frame so that a fair amount of grayscale is visible. Then in the B&C box click “apply”.
8. Select paintbrush by double clicking it. This will not only switch to paint brush but it will allow you to change the thickness of the brush as well. Adjust brush thickness to 1 pixel.
9. Use zoom in and zoom out to manipulate the frame. This can be accomplished by pressing the magnifying glass and then by left clicking to zoom in and right clicking to zoom out.
10. In order to trace the cells well, zoom in really far in on the edge of the cell. Trace the cell by having the brush edge paint over the very edge of the cell.
11. Save the frame by clicking “file”, “save as”, “.tif” then choose where to save it.
12. To make a mask, you must first duplicate your images. To do this, click on image, then click duplicate, and check the box that says “duplicate stack”.
13. Go to Image, then Type, and select 8- Bit.
14. Next, click on image, then click adjust, then click threshold.
15. In the first dropdown menu, make sure that “default” is selected and in the second dropdown menu, make sure that “red” is selected.
16. Adjust the two scrollbars until you see a red outline around each cell with as little red within the cell as possible. Click on apply.
17. Next click on process, then click binary, then click fill holes. (NOTE: Real interior holes will be lost)

18. Finally, if necessary, click on analyze, then on analyze particles to get rid of any spots that are not cells. You will have to change the setting to 2100-infinity and you will need to select "Show Masks" in the dropdown menu. Duplicate this image.

Appendix 2: An ImageJ macro code for computing GT areas, AD areas, FPs and FNs in fluorescent microscopy images

Cell culture methods: A-10 rat smooth muscle cells (ATCC, Manassas, VA) were maintained in Dulbecco's Modified Eagles Medium (DMEM/10 %FBS, Mediatech, Herndon, VA) supplemented with glutamine, non-essential amino acids and occasionally penicillin/streptomycin (Invitrogen, Carlsbad CA) in 5 % CO2 at 37 °C**.** For the experiment, the cell lines were seeded at 800, in 3-wells of a 6-well tissue culture treated polystyrene plate (353046, BD Falcon, Franklin Lakes, NJ) in maintenance media, and placed in the incubator for approximately 20 hours. The media was removed; the cells were rinsed with PBS and fixed for 3 h with 1 % (v/v) formaldehyde in PBS at 25 °C. The cells were stained with PBS containing 0.02 % (v/v) TritonX-100 (Sigma, St. Louis, MO), 0.5 µg/mL TxRed c2 maleimide (Invitrogen) (5 mg/mL in DMSO stock), 1.5 µg/mL DAPI **)** (Sigma) (1 mg/mL in DMSO stock) for 4 hours, rinsed with PBS, PBS containing 1 % BSA and PBS, sequentially. Fixed and stained cells were covered with PBS, stored at 4 °C, and imaged within two days.

Automated Fluorescence Microscopy Imaging details: Fluorescence images of fixed and stained cells were acquired with an Olympus IX71 inverted microscope (Center Valley, PA) equipped with an automated stage (Ludl, Hawthorne, NY), automated filter wheels (Ludl), a Xe arc lamp fluorescence excitation source, a 10 x ApoPlan 0.4 NA objective (Olympus), and a CoolSNAP HQ CCD camera (Roper Scientific, Tucson, AZ). the TxRed stained cells imaged using a 555 nm notch excitation (PN# S555_25x) and a 630 nm notch emission filter (PN#S630_60m) and a custom coated multipass dichroic beam splitter (PN#51019+400DCLP) matched to the excitation and emission filters (Chroma Technologies, Brattleboro, VT).

ImageJ macro code for computing the GT area, AD area, FPs and FNs:

```
//list of thresholds
num_thresholds=17;
threshold_type=newArray(num_thresholds);
threshold_type[0]="Default";
threshold_type[1]="Huang";
threshold_type[2]="Intermodes";
threshold_type[3]="IsoData";
threshold_type[4]="IJ_IsoData";
threshold_type[5]="Li";
threshold_type[6]="MaxEntropy";
threshold_type[7]="MinError";
threshold_type[8]="MinError";
threshold_type[9]="Minimum";
threshold_type[10]="Moments";
threshold_type[11]="Otsu";
```

```
threshold_type[12]="Percentile";
threshold_type[13]="RenyiEntropy";
threshold_type[14]="Shanbhag";
threshold_type[15]="Triangle";
threshold_type[16]="Yen";

//where to write the data files
save_path=getDirectory("Choose a directory to save the analysis results");

//NOTE: make sure ROI manager is cleared and closed
waitForUser("Select the reference mask images...");
referenceID=getImageID();
waitForUser("Select the images to segment...");
testID=getImageID();

//make blank image to use for calculating AREA of algorithm, FP, and FN
selectImage(testID);
wt=getWidth();
ht=getHeight();
run("Select None");
run("Duplicate...", "title=test_space");
run("8-bit");
run("Multiply...", "value=0");
run("Canvas Size...", "width="+wt+2+" height="+ht+2+" position=Center zero");

//set-up ROIs based on reference image
selectImage(referenceID);
run("Analyze Particles...", "size=500-Infinity circularity=0.00-1.00 show=Nothing exclude add
stack");
roiManager("Show None");

cells= roiManager("count");

//setting up arrays for reference, false positive, false negative, and test areas
ref_area = newArray(cells);
FN_area = newArray(cells);
FN_area2 = newArray(cells);
test_area = newArray(cells);
test_area2 = newArray(cells);
FP_area  = newArray(cells);
FP_area2  = newArray(cells);
TP_area = newArray(cells);
TP_area2 = newArray(cells);
num_particles = newArray(cells);

//identify corresponding cells in test image after threshold
```

```
run("Set Measurements...", "area centroid redirect=None decimal=3");
for (h=0; h<num_thresholds; h++)     {
//make duplicate image of test image to set threshold and make mask image
                    selectImage(testID);
                    run("Duplicate...", "title=stack_analysis duplicate range=1-50");
                    setAutoThreshold(threshold_type[h]+" dark");//autothreshold test image
                    getThreshold(lower, upper);
                    run("Convert to Mask", " black");
                    run("Canvas Size...", "width="+wt+2+" height="+ht+2+" position=Center zero");
          for (i=0; i<cells; i++)  {
//get reference area
                    selectImage(referenceID);
                    roiManager("Select", i);
                    slice_n=getSliceNumber();
                    getStatistics(area, mean);
                    ref_area[i] = area*mean/255;
//find all particles under the reference mask
                    TP = 0;//true positives area
                    store_test_area = 0;
                    run("Clear Results");
                    selectImage(testID);
                    setThreshold(lower, upper);//resets autothreshold from earlier
                    roiManager("Select", i);
                    run("Analyze Particles...", "size=0-Infinity circularity=0.00-1.00 show=Nothing
display");
//paste each detected fragments into the test space
                    for (j=0; j<nResults; j++)       {
                              x = getResult("X", j);
                              y = getResult("Y", j);
                              TP = TP + getResult("Area", j);
                              selectWindow("stack_analysis");
                              setSlice(slice_n);
                              run("Select None");
                              doWand(x+1, y+1);
                              roiManager("Add");
                              selectWindow("test_space");
                              roiManager("Select", cells);
                              run("Fill", "slice");
                              roiManager("Delete");
                    }
          getStatistics(area, mean, min, max, std);
          test_area[i] = store_test_area;
          test_area2[i] = mean*area/255;
          run("Select All");
          run("Translate...", "x=-1 y=-1 interpolation=None");
          roiManager("Select", i);
```

```
		getStatistics(area, mean, min, max, std);
		FN_area[i] = ref_area[i]-TP;
		FN_area2[i] = area-(area*mean/255);
		TP_area[i] = TP;
		TP_area2[i] = area*mean/255;
		run("Multiply...", "value=0");
		run("Select None");
		getStatistics(area, mean, min, max, std);

		FP_area[i] = test_area[i]-TP;
		FP_area2[i] = area*mean/255;
		num_particles[i] = nResults;
//remove detected fragments and clear results
		run("Clear Results");
		run("Multiply...", "value=0");
		}

//reset threshold and close the mask image made earlier
selectImage(testID);
resetThreshold;
selectWindow("stack_analysis");
close();

//write results to file
f=File.open(save_path+threshold_type[h]+".txt");
print(f, "GT pixels, algorithm, FP, FN, TP, num_particles");
for (i=0; i<cells; i++)  {
		print(f, ref_area[i]+", "+test_area2[i]+", "+FP_area2[i]+", "+FN_area2[i]+",
"+TP_area2[i]+", "+num_particles[i]);
		}
		File.close(f);
}

//print results table
//run("Clear Results");

//for (i=0; i<cells; i++)		{
//		setResult("GT pixels",i,ref_area[i]);  //print results to table
//		setResult("Algorithm",i,test_area[i]);
//		setResult("FP",i,FP_area[i]);
//		setResult("FN",i,FN_area[i]);
//		setResult("TP",i,TP_area[i]);
//		}


//updateResults(); //show results
```

//selectWindow("Results"); //bring results to front - this data should be pasted into a spread sheet program

References

1. K. Wu, D. Gauthier and M.D. Levine, Live cell image segmentation, IEEE Trans. on Biomedical Engineering 42 (1), 1-12 (1995).
2. J.C. Wu, A.F. Martin and R.N. Kacker, Measures, uncertainties, and significance test in operational ROC analysis, J. Res. Natl. Inst. Stand. Technol. 116 (1), 517-537 (2011).
3. J.C. Wu, A.F. Martin, C.S. Greenberg and R.N. Kacker, Data dependency on measurement uncertainties in speaker recognition evaluation, in Active and Passive Signatures III, Proc. SPIE 8382, 83820D (2012).
4. P. Jaccard, Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines, Bulletin de la Société Vaudoise des Sciences Naturelles 37, 241-272 (1901).
5. W.M. Rand, Objective criteria for the evaluation of clustering methods, Journal of the American Statistical Association, 66 (336), 846–850 (1971).
6. A.P. Zijdenbos, B. M. Dawant, R.A. Margolin and A.C. Palmer, Morphometric Analysis of White Matter Lesions in MR Images: Method and Validation, IEEE Trans. Medical Imaging, 13 (4), 716- 724 (1994).
7. F. Yang, M.A. Mackey, F. Ianzini, G. Gallardo and M. Sonka, Cell Segmentation, Tracking, and Mitosis Detection Using Temporal Context, in J. Duncan and G. Gerig (Eds.): MICCAI 2005, LNCS 3749, 302-309 (2005).
8. L.P. Coelho, A. Shariff and R.F. Murphy, Nuclear segmentation in microscope cell images: A hand-segmented dataset and comparison of algorithms, in Proc IEEE Int Symp Biomed Imaging, 518–521 (2009).
9. J. Strijbos, R. Martens, F. Prins and W. Jochems, Content analysis: What are they talking about? Computers & Education, 46, 29–48 (2006).
10. "NIST Semantics for Biological Data Resource: Cell Image Database", National Institute of Standards and Technology, http://sbd.nist.gov/, (2012).
11. Rasband, W.S., ImageJ, U. S. National Institutes of Health, Bethesda, Maryland, USA, http://rsbweb.nih.gov/ij/, 1997-2008.
12. D.J. Hand, Construction and assessment of classification rules, John Wiley & Sons, New York, (1997).
13. B.L. van der Waerden, Mathematical Statistics, Springer, Berlin, (1969).