Identicyte: Simple red blood cell identification software

Guillaume F.G. Garnier, Clare A. Manderson, Saveen Giri, Gil Garnier

Bioresource Processing Research Institute Australia (BioPRIA), Department of Chemical Engineering, Monash University, Clayton VIC 3800, Australia

Abstract
Counting and identifying red blood cells (RBCs) by shape is an important step in diagnosing several diseases and disorders. This is traditionally done by hand (a tedious and time consuming process) or with specialized equipment. There exist several tools to count and classify cells in an image automatically; however, they often sacrifice simplicity in favour of having a wide range of applications. In this paper, we present Identicyte, a program specifically designed to quickly count and identify RBCs from a series of microscope images as simply as possible.

© 2019 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Motivation and significance
The identification and classification of red blood cell (RBC) shapes is an important problem for diagnostic, transfusion and...
biomedical applications. RBC shape can help to diagnose diseases and disorders such as leukaemia, sickle cell anaemia and malaria [6]. Cells are traditionally counted manually, which is tedious, time consuming and adds a factor of human inconsistency. As a result, many methods of automating this process have been developed. Some of these rely on expensive, specialized instruments [7,8], which can be impractical in many situations. A more affordable method is to use software to analyse images taken by a simple optical microscope. Here, we present IdentiCyte as a simple piece of software for automated counting and identifying of cells based on shape.

IdentiCyte was developed as part of a project to quantify RBC (erythrocyte) shape change caused by their suspension in various solutions. Unstained cells suspended in their solutions were imaged directly by an optical microscope for examination. By initially classifying some example cell shapes into six distinct categories (Fig. 1) for the library, IdentiCyte was able to efficiently and reliably provide cell population statistics for ten of thousands of cells, from hundreds of optical microscope images. When identified manually, the same cell may be classified differently by different people. Also, contrast-enhancing dyes could not be added to the cells to allow for easy visualization, as doing this could alter their native shape. This leads to the cells often being quite faint in the microscopy image, making the different shapes hard to distinguish.

We created IdentiCyte because we struggled to find user friendly software to identify and count the cell types in our microscopy images. Several previous publications [6,9–11] have described their methods and results without making their software readily available. Several pieces of software have been developed and published for public use. The first among these is BlobFinder [12] which was also developed as a simple way to count cells. BlobFinder only counts cells without identifying them, however. Further, Blobfinder was developed specifically for fluorescent image microscopy. Currently, there is a beta version of Blobfinder that is built to work with bright field images.

Another software of interest used to analyse cells is CellProfiler [13]. It is a very powerful tool that applies image processing steps to images in a pipeline. While CellProfiler itself only applies image processing, there is a companion software, CellProfiler Analyst [14], which can use machine learning techniques to count and identify cells. CellProfiler is very powerful, especially when paired with CellProfiler Analyst, but its power comes at the cost of simplicity. There are many modules, each of which has several options to parse. The complexity is only increased by the fact that two separate pieces of software need to be run to get complete results.

IdentiCyte has been designed to automatically identify RBCs in a batch of images with as little difficulty as possible for the user. The analysis process can be broadly split into two main phases: the first, is building a library of examples of cell identifications; and the second, is running the program on the images to be identified. A library is created by placing examples of cells from each class in the folder corresponding to that category. We recommend using the Extract Cells function provided in IdentiCyte to obtain these as it adheres to the requirements of the program for identification. Once a database of cell images has been placed in the correct folders of the library, the library can be compiled using the Compile Library button in the Library tab. In the compilation process, all the images in the library folders are read and transformed, and the results are saved to a LibraryInfo file. The file contains a matrix of orthogonal vectors which represent all the cells in the Library, as well as a list of the category to which each cell belongs. Extracted cells are in 3-bit grey scale format by default, but the bit depth can be changed by the user.

The recognition process is also separated into two distinct parts: detection and recognition. The detection phase uses image processing techniques to determine the location of the cells in each image. As shown in Fig. 2 the detection begins by taking an image of cells (2.a) and extracting a single colour channel of the image (2.b). To this an automated binary threshold (using either Otsu’s [18] or Triangle [19] method) (2.c) is applied. The resulting binary image then has its holes filled (2.d), which helps in calculating the area and ensures that a single cell is not incorrectly segmented in the next step. This image is then segmented with the watershed method [20] to separate multiple cells that are seen as one dark blob (2.e) so they can be identified individually. This process renders a white image with black spots, each of which can have its area and centre calculated (2.f). The measured area is used to discard spots that are too small, which is useful for removing debris from the image.

The recognition phase, depicted in Fig. 3, takes the cells that have been found through detection, converts them to 3-bit grey

We instead choose to identify RBCs using eigenfaces, a method proposed by Turk et al. [17] originally designed for facial recognition which has been augmented by a novel confidence metric. This method uses principal component analysis to extract features from a library of pre-identified cells. These features can be used to compare new images to those in the library and determine which is most similar.

One key limitation of this method is that the use of a space means that the images in the library must all have the same number of pixels (and for simplicity, the same dimensions). This means that the cells to be analysed should all be approximately the same size.

This method offers the advantage that images can be compared with few key features (or principal components). This is not only a bonus for computational complexity, it also may improve accuracy. Also the number of cells manually identified and used in the library can be comparatively small when compared to the number of cells that can be automatically identified. In our project for example, we constructed a library containing merely a few hundred of cells to analyse hundreds of thousands of cells in images.

IdentiCyte is not limited to a single operating system, but it can be run from a Windows executable and analysis can be done by clicking a single button. IdentiCyte allows for new categories of RBCs or other cells to be defined by the user.

2. Software description

2.1. Software architecture

IdentiCyte has been designed to automatically identify RBCs in a batch of images with as little difficulty as possible for the user. The analysis process can be broadly split into two main phases: the first, is building a library of examples of cell identifications; and the second, is running the program on the images to be identified. A library is created by placing examples of cells from each class in the folder corresponding to that category. We recommend using the Extract Cells function provided in IdentiCyte to obtain these as it adheres to the requirements of the program for identification. Once a database of cell images has been placed in the correct folders of the library, the library can be compiled using the Compile Library button in the Library tab. In the compilation process, all the images in the library folders are read and transformed, and the results are saved to a LibraryInfo file. The file contains a matrix of orthogonal vectors which represent all the cells in the Library, as well as a list of the category to which each cell belongs. Extracted cells are in 3-bit grey scale format by default, but the bit depth can be changed by the user.

The recognition process is also separated into two distinct parts: detection and recognition. The detection phase uses image processing techniques to determine the location of the cells in each image. As shown in Fig. 2 the detection begins by taking an image of cells (2.a) and extracting a single colour channel of the image (2.b). To this an automated binary threshold (using either Otsu’s [18] or Triangle [19] method) (2.c) is applied. The resulting binary image then has its holes filled (2.d), which helps in calculating the area and ensures that a single cell is not incorrectly segmented in the next step. This image is then segmented with the watershed method [20] to separate multiple cells that are seen as one dark blob (2.e) so they can be identified individually. This process renders a white image with black spots, each of which can have its area and centre calculated (2.f). The measured area is used to discard spots that are too small, which is useful for removing debris from the image.

The recognition phase, depicted in Fig. 3, takes the cells that have been found through detection, converts them to 3-bit grey
scale and compares them to the cells in the compiled library file. This is done by arranging the pixel intensities of the unidentified cell into a vector, \( \mathbf{x} = (x_1, x_2, \ldots, x_n) \) and projecting it onto the space defined by the library matrix which is comprised of vectors representing each cell \( i \) in the library \( \mathbf{y}_i = (y_{i1}, y_{i2}, \ldots, y_{in}) \). One way to define similarity is to use the Euclidean distance between these two vectors. The Euclidean distance, \( d_i \), between \( \mathbf{x} \) and every \( \mathbf{y}_i \) can be found by

\[
d_i = \sqrt{\sum_{j=1}^{n} (x_j - y_{ij})^2}
\]

These distances are then ranked from nearest to furthest. It would be easy to give the cell being analysed the same classification as the cell to which it is nearest in the library. Unfortunately, the nearest cell does not necessarily represent the best category. Consider the simplified example in Table 1 which shows sample distances from analysing a cell. Here, Categories 1, 2 and 3 are distinct cell shapes. By naively selecting the nearest category, the target cell would be considered Category 1. However, there are two Category 2 cells in the library that have similarly short distances from the unknown cell. So, it may be unclear whether it is actually Category 2 or Category 1.

We present a new metric, confidence (Eq. (2)), which considers rank and distance to more accurately identify the cell.

\[
\text{confidence}(\text{shape}) = \sum_{i=1}^{10} \left( \frac{1}{\text{rank}_i} \times \frac{1}{\text{distance}_i} \right)
\]

For each of a small number of nearest neighbours, the confidence is calculated. We then calculate which of the categories has a greater representation by calculating a percentage. Those with the greatest percentage above a user defined threshold are automatically classified. This is necessary because certain aspects of cell shape can vary significantly within the same category, and the distances between a cell and the nearest match in the library can differ greatly. For the example above, the confidence percentages (shown in Table 2) indicate that the cell is in Category 2.

The presented confidence metric is moderately sensitive to the number of nearest neighbours chosen. The intent was to use the rank of the cell to discount the contribution of cells that are not as close as others, even if they have a relatively small distance. This was done in order to further separate categories that represent
Fig. 3. The process of recognition showing all options for user verification.

Table 2
The confidences for each type of cell from the previous table calculated using Eq. (2). Using this method the cell is more accurately identified as Category 2.

<table>
<thead>
<tr>
<th>Confidence score</th>
<th>Confidence percentage</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0500</td>
<td>42.7%</td>
<td>Category 1</td>
</tr>
<tr>
<td>0.0628</td>
<td>53.7%</td>
<td>Category 2</td>
</tr>
<tr>
<td>0.0042</td>
<td>3.6%</td>
<td>Category 3</td>
</tr>
</tbody>
</table>

similar continuous states, however the option was not as effective as had been anticipated. In general, as the number of neighbours considered is increased, the number of cells that are definitively classified goes down. This is because many distant cells reduce the confidence and it becomes more difficult for a single cell type to meet the threshold required to automatically classify a cell.

Also, too few can lead to cells being miscategorised, especially for cell types which have a lot of variation in their appearance like echinocytic cells. In our experiments, we found that choosing the nearest 10 cells provides a good middle ground for balancing these two properties.

The cell’s classification, confidence and location are saved to a file which can be used to review of cell identifications. When all the cells in each image within a set have been identified, statistics from the analysis are saved to an Excel document.
2.2. Software functionalities

The IdentiCyte user interface (pictured in Fig. 4) is made up of four tabs: the Analysis tab, the Review tab, the Library tab and the Options tab; a display window, and a Cancel button. The display window will show messages from the software relating to its stage in analysis. The Analysis tab analyses images by finding cells within them and identifying them into user-defined categories. There is also an option for users to identify cells with a low confidence manually as depicted in Fig. 3. Cells identified in this way will be added to the library, in a way 'teaching' the software to identify cells better.

IdentiCyte also has the capability to extract cells for the library. Once these cells have been manually sorted into their category folders, they can then be compiled. All functions pertaining to the library can be found in the Library tab.

In the review tab, images that have already been identified can be viewed with labels over cells to show the identifications. This reads in the positions of the cells and the result of analysis and shows the image with colour coded labels (Fig. 5) that state the identification of the cells and the confidence of the identification. These labelled images can be saved by pressing the ‘s’ key when they are displayed for presentation or display outside of the software.

Finally, users are also given the ability to change several key variables in the operation of IdentiCyte. Among these are the colour channel selected in the images, the thresholding method, illumination, the bit depth of cell representations in the library, the width of the window in which cells are extracted, minimum pixel area for an object to be considered a cell, and the confidence percentage a cell must have to be definitely put in a cell.

The output Excel file comprises two sheets: a summary tab which gives an overview of the entire batch of images analysed, and a per image breakdown, which shows how many cells are in each image, and statistics on category variation. The summary sheet displays the total number of cells analysed, the number of cells of each type in the batch, and the average confidence across all images. A summary of all the settings used for the sample has also been included. The second sheet contains the number of each type of cell in an image as well as in the whole batch, given as a percentage of the total number of cells in that image. The average confidence score of all cells in each image is also displayed.

3. Illustrative examples

Much effort has been expended in the development of IdentiCyte in ensuring that it is as simple as possible to analyse cells and review the results. Here we will give an example of how to analyse a batch of images with the included library and view the results of analysis.

IdentiCyte opens to the “Analysis” tab by default, so the first step in analysing a batch of images is to select the folder containing them. This can be done by pressing the ‘...’ button next to the text box labelled “Input Folder”. This brings up a folder selection window. Once this has been selected, the next step is to click the “Identify Cells” button. This will begin the analysis, as seen in Fig. 4. The output box will display what the program is currently doing. First, it will display the path to the folder containing the images to be analysed, and indicating that the Identification is starting. As IdentiCyte analyses images it will display the name of the image it is currently analysing. After all images have been analysed, the files containing the results will be completed during which “Outputting to Excel” will be displayed. Upon completion of the output, the program will display “Done”. Once the analysis has finished, the results can be reviewed by switching to the “Review” tab and clicking the “Review” button. This brings up a window (Fig. 5) that shows the identification of each cell in an image. The overall statistics from the analysis are also provided in an Excel document in the folder containing the analysed images.

4. Impact

IdentiCyte is a simple and robust program to facilitate the counting of cells and identification based on shape in image...
Fig. 5. The review option displays the cell identification with corresponding confidences as colour coded labels over the users original image data.

microscopy. IdentiCyte is designed to be used to streamline the process of analysing cells. As a direct consequence of the fact that it is an automated cell counter, a project which uses IdentiCyte will not need a human researcher to count cells themselves. Related to this is another advantage of IdentiCyte: it counts cells quickly, taking just under five and a half minutes for more than 1700 cells on an Intel Core i7-7700HQ with 16 GB of RAM. This means that the results of a sample are available more quickly than if they were hand counted.

When humans classify cells, they must decide for each cell which type it is. As the number of cells increases so too does the amount of time it takes to count the cells, after enough time the counter is likely to become fatigued which leads to an inconsistency of accuracy in human counting. Further, when categories of cells are similar, different people may disagree in which category a cell should be. This illustrates the key advantages of IdentiCyte, its consistency, objectivity and speed. As IdentiCyte is a program, when it is given the same input data, it will produce the same output, and while there will always be cells categorized incorrectly, this inaccuracy will be a constant factor which may be easily be accounted for when analysing results. When constructing the library, fewer cells need to be identified relative to the total amount, which also allows for library construction to be done collaboratively. This helps to reduce subjectivity in classification as multiple people can verify a relatively small amount of cell classifications without succumbing to the effects of fatigue.

IdentiCyte has been used in research determining cell shape in relation to solution properties, where it was used in place of a flow cytometer, giving advantages; the first is that much time was saved in not having to operate the device, the second is that the flow cytometer gives limited information on cell shape in the form of light scatter. Finally, the flow cytometer requires the cells to be stained, which can move the cells out of their native shape. As IdentiCyte is free, it can also be used in situations where purchasing a flow cytometer or other equipment is not a viable option. Our emphasis on simplicity also allows IdentiCyte to be used without formal training or knowledge of coding.

While IdentiCyte has been specifically designed to identify and count RBCs, it could be expanded to identify other types of cells by creating a new library.

5. Conclusions

With IdentiCyte, we present a simple interface to identify the shape and count large quantities of cells captured with a microscope. This emphasis on usability allows for the benefits of automated cell identification without the need to spend much time learning the intricacies and features of specialized software, or the expense and time required to operate a dedicated machine.

Acknowledgements

This work was supported by the Australian Research Council (LP110200973) and Haemokinesis. We would like to thank the Australian Red Cross Blood Bank for providing blood samples, Samuel Li and Daniel Ly for contributing test images.

Conflict of interest

The authors declare that there is no conflict of interest.

References


