QuickCount[®]: a novel automated software for rapid cell detection and quantification

Kai Hung Tiong^{*1}, Jit Kang Chang^{†,2,3}, Dharini Pathmanathan⁴, Muhammad Zaki Hidayatullah Fadlullah¹, Pei San Yee¹, Chee Sun Liew^{2,3,5}, Zainal Ariff Abdul Rahman⁶, Kheng Ling Beh⁷ & Sok Ching Cheong^{*1,6}

ABSTRACT

We describe a novel automated cell detection and counting software, QuickCount[®] (QC), designed for rapid quantification of cells. The Bland–Altman plot and intraclass correlation coefficient (ICC) analyses demonstrated strong agreement between cell counts from QC to manual counts (mean and SD: -3.3 ± 4.5 ; ICC = 0.95). QC has higher recall in comparison to ImageJ_{auto}, CellProfiler and CellC and the precision of QC, ImageJ_{auto}, CellProfiler and CellC are high and comparable. QC can precisely delineate and count single cells from images of different cell densities with precision and recall above 0.9. QC is unique as it is equipped with real-time preview while optimizing the parameters for accurate cell count and needs minimum hands-on time where hundreds of images can be analyzed automatically in a matter of milliseconds. In conclusion, QC offers a rapid, accurate and versatile solution for large-scale cell quantification and addresses the challenges often faced in cell biology research.

¹Head & Neck Cancer Research Team, Cancer Research Malaysia, Subang Jaya, Selangor, Malaysia; ²Data-Intensive Computing Centre, University of Malaya, Kuala Lumpur, Malaysia; ³Department of Computer System & Technology, Faculty of Computer Science & Information Technology, University of Malaya, Kuala Lumpur, Malaysia; ⁴Institute of Mathematical Sciences, University of Malaya, Kuala Lumpur, Malaysia; ⁵Centre for Data Analytics, University of Malaya, Kuala Lumpur, Malaysia; ⁶Department of Oral & Maxillofacial Clinical Sciences, Faculty of Dentistry, University of Malaya, Kuala Lumpur, Malaysia; ⁷ViTrox Technologies Sdn. Bhd., Bayan Lepas, Penang, Malaysia; ^{*}Author for correspondence: sokching.cheong@cancerresearch.my

[†]Both authors contributed equally to this work

METHOD SUMMARY

QuickCount[®] is a novel software for rapid detection and quantitation of fluorescently labelled fixed cells. Optimal parameters for cell detection can be adjusted using a real-time preview window. When uploaded by the user, a large number of images can be analyzed simultaneously and the data are tabulated automatically.

KEYWORDS:

accurate • automation • cell count • cell quantification • scalable • time-efficient • user-friendly

INTRODUCTION

A change in cell density is often used in cancer research as an indicator to assess antiproliferative or cytotoxic activities in response to potential therapeutic agents [1]. As DNA replication precedes cell division, one of the most direct ways of measuring cell proliferation is by measuring DNA synthesis. For example, a nonradioactive approach for measuring DNA synthesis is by detecting fluorescent azides-tagged EdU, a nucleoside analog of thymidine that is incorporated into DNA as it is being synthesized in cells [2]. In this way, the number of replicating cells can be determined from red fluorescent-labelled stained cell nuclei, relative to the total number of cell nuclei stained in blue by DAPI (4',6-diamidino-2-phenylindole).

Manual cell counting by examination of a microscopic field in real-time or counting of cells through images taken of different microscopic fields is regarded as the gold-standard for determining cell numbers in *in vitro* assays [3]. Typically, this has relied on open-source tools such as ImageJ, where cells can be manually marked as they are counted (ImageJ_{man}). The main advantage of this approach is the ability to accurately identify which entities or cells should be counted. However as this process is laborious and prone to human error, particularly when a large number of cell images

are being analyzed [4], there is a critical need for the development of an automation step that can improve accuracy, speed and reproducibility. At present, several pieces of commercially available automated cell counting software are available; however, these are typically embedded within image processing suites that are costly, or the use of automation relies on the ability of users to code [5–8].

Here we describe the development of an automated, adaptable and user-friendly cell counting tool, QuickCount[®] (QC), for the analysis of a large number of images in an efficient and precise manner. We determined the versatility of QC in counting cells of different lineages compared with currently available software, and manual counts facilitated by ImageJ (ImageJ_{man}) were used as the ground truth.

MATERIALS & METHODS

Development of QC, a cell detection & counting tool

QC is a software built on a cell-counting algorithm written in C++ language. The algorithm consists of six major processes (Figure 1A).

1) Color channel extraction: image colors will be extracted according to the respective red, green and blue (RGB) color channels based on image samples' filename. For instance, the red signal of Alexa 647 is first extracted from the red channel, as shown in Figure 1b. The extraction of a single channel from the RGB components in the image reduces the original pixel details from 24-bit to 8-bit and renders the resulting image in grayscale.

2) Brightness and contrast adjustment: to increase the distinctiveness of the cell nuclei, the brightness and contrast of boundaries between cell nuclei and background will be adjusted by multiplying the intensity value of each pixel v with n , where $n \geq 1$ and product values of v and n are between 0 to 255. By default, a value of $n = 7$ is applied where ►

► most pixels that have higher intensity will receive a higher boost in intensity that renders them much brighter than the pixels that have lower intensity (Figure 1C). An example of brightness and contrast adjustment can be found in Figure S1.

3) Binary thresholding: this process is performed to remove noise or blotches present at exceptionally low pixel intensity (Figure 1D). As intensity and contrast varies from one image to another, a default threshold value of 240 is applied to all images (higher threshold value will be selected if the image is brighter, and vice versa). An appropriate threshold value could also be adjusted through real-time preview on the graphical user interface. When applying binary threshold on an image, image pixels with an intensity value greater than or equal to the threshold value will be set to the highest value, 254, while any pixel intensity values less than the threshold value will be set to 0. The resulting image will consist of multiple 0 and 255 values across the whole image, which potentially reduces computational workload of the CPU as the image is now treated like a 1-bit image in black and white.

4) Gaussian blurring: we applied Gaussian blur on the image to smoothen and reduce noise by preserving only the intensity, contrast and shape of nuclei in the images (Figure 1E).

5) Morphological opening: this is derived from the fundamental operation of image erosion and dilation. While erosion can be used to effectively eliminate small clumps of undesirable foreground pixels, it affects all regions of foreground pixels indiscriminately. Morphological opening gets around this disadvantage by performing both erosion and dilation on the image. Elliptical and circular shapes were chosen as the kernel shape for the opening process as most of the cell nuclei are oval shaped (Figure 1F). By adjusting the size of the kernel, we can determine how much 'salt-noise' in the image to be removed.

6) Cell nuclei approximation: following image processing, the total number of cells is estimated by observing the change of pixel intensities across the image to detect all contours found within each image. Cell centroid of each detected nuclei is calculated and estimated by averaging all the points in each contour to find the midpoint of the nuclei. The calculated cell centroid will be

marked as a detected cell in the preview window.

Evaluation of QC performance

Cell culture & Click-iT™ EdU assay

U2OS (human osteosarcoma), NIH3T3 (murine fibroblast), A431 (human squamous cell carcinoma of the skin) and A549 (human squamous cell carcinoma of the lung) cells were obtained from ATCC and cultured in Dulbecco's Modified Eagle Medium (DMEM; GIBCO, Carlsbad, CA, USA) containing high glucose, while MCF7 (human breast cancer), HT29 and HCT116 (human colorectal cancer), and 4T1 (murine breast cancer) cells were grown in Roswell Park Memorial Institute (RPMI) 1640 media (GIBCO, CA, USA). ORL-150, a human oral squamous cell carcinoma cell line was cultured as described previously [9]. The Click-iT™ 5-ethynyl-2'-deoxyuridine (EdU) assay kit (Invitrogen, CA, USA) was used to measure cell proliferation as described previously [9]. Briefly, cells were seeded on cover slips overnight prior to the addition of 10 μ M EdU and cells were incubated between 2 and 6 h as optimized previously, to allow the incorporation of EdU. Cells were then fixed with 3.7% formaldehyde in phosphate-buffered saline (PBS) and washed with 3% BSA in PBS prior to permeabilization by 0.5% Triton-X in PBS and stained with fluorescent dyes, Alexa 647 and DAPI, in the dark for 30 min each. Fluorescent-stained nuclei images were captured from mounted glass slides and the percentage of proliferating cells were expressed as the percentage of EdU-positive cells stained in red by Alexa647 over the total number of cells stained in blue by DAPI. To evaluate the performance of QC at difference cell densities we used images from experiments where cells were treated with a CDK-1 inhibitor, RO3306. Cells were either treated with DMSO control (0.1% DMSO) or RO3306 at two drug concentrations (5 and 10 μ M) for 24 h and the Click-iT™ EdU assay was performed as described above.

Image capturing & processing

Images from the experiments as described above were captured using an Olympus IX71 microscope (Olympus, Japan), at 10x magnification. The excitation and emission wavelengths for Alexa 647 were 650 nm and 667 nm, respectively, and for DAPI they were 360–370 nm and 420 nm, respectively. All

images were labeled according to a standard format 'Treatment Concentration (in μ M) Fluorescent dye (Alexa 647 or DAPI) exposure time (in ms) image number (n)' and saved in PNG format. Cell counting was performed by using: 1) Manual cell counting with the assistance of the ImageJ program (ImageJ_{man}) (v1.49u, National Institute of Health, MD, USA), where cell counting was performed by clicking manually on each cell while a counter records each click as one count; 2) Automated cell counting using the macro-automated function of the ImageJ program (ImageJ_{auto}) (v1.49u, National Institute of Health) [10,11]; 3) CellC (v1.2, Tampere University of Technology) [12], a cell-counting software with user-specified parameters such as intensity threshold and cell clusters division; 4) CellProfiler [13] (v2.2.0 rev ac0529e) where the 'Identify primary objects' module available by default in the software was used; and 5) QC as described below. All true-positive, false-positive and false-negative were validated and tabulated from the generated image outputs from QC, ImageJ_{auto}, CellC and CellProfiler.

Image presentations

All images are represented in computer vision as multidimensional arrays containing pixel intensity values. The first two dimensions are typically meant to be spatial, while others are unspecified and can contain for instance colour channels, z-slices, replicates, time points or combinations of different properties. Image representation is dissociated from rendering, and multidimensional arrays can be displayed as animated sequences of images in greyscale or colour mode. As matrices, images can be manipulated with algebraic operators such as sum, product, comparison or convolution. QC can be used on images in different formats as OpenCV, an open-source computer vision library that is used to read and process the images. Files that can be processed include Windows bitmaps (*.bmp), JPEG files (*.jpeg, *.jpg, *.jpe), JPEG 2000 files (*.jp2), Portable Network Graphics (*.png), Portable image format (*.pbm, *.pgm, *.ppm), Sun rasters (*.sr, *.ras), and TIFF files (*.tiff, *.tif).

Figure 1. The QuickCount® (QC) workflow and visualization of image processing prior to detection and cell counting. (A) Sample images are loaded into QC and the program automatically determines the color channel and extracts the channel based on image type. (B) An example of Alexa 647-stained nuclei is shown. (C) Brightness and contrast adjustments, (D) binary thresholding, (E) Gaussian smoothing and (F) morphological opening, were applied to images prior to analysis.

► Comparison analyses between ImageJ_{man}, ImageJ_{auto}, CellC, CellProfiler & QC

To evaluate the performance of QC in identifying and counting cells, the software output counts were compared with the gold standard of ImageJ_{man}. As QC is designed to be automated, we also compared its performance with the macro-automated functions of ImageJ (ImageJ_{auto}), CellC and CellProfiler. The workflows in processing the images and using each of these tools are described in Figure 2A & Figure 2B. For ImageJ_{auto}, common pre-processing steps such as 32-bits conversion and Gaussian smoothing were performed to ensure accurate cell detection by the macro. All pre-processing steps were recorded in a macro script to

reduce human intervention and to standardize timing for each step. In QC, sample of images were used to adjust for optimum settings through the parameter sliders and ideal settings were identified through observing the real-time display window where markers are shown on counted cells. These uniformed settings were then applied for all subsequent images. For CellC, an image was loaded into 'Figure 1 container' in the software before parameters such as automatic intensity threshold, cell clusters division rate, cluster division algorithm, or removal of objects size were determined, and this was followed by clicking on 'analysis' to obtain a cell count result. The analysis was repeated several times with different parameters and this was

checked against the markers on the counted cells as described above to ensure optimal parameters were selected. With optimal parameters, the remaining images were processed and counted by using the 'batch processing' feature available in CellC. For CellProfiler, several modules were imported into the pipeline and these were the 'Color-ToGray', 'IdentifyPrimaryObjects' and 'ExportToSpreadSheet'. The Alexa647 and DAPI images were loaded separately into the software as only one colour channel can be selected for each analysis.

The performance of these tools to delineate single cells for count across different cell lineages was evaluated by comparing cell counts from each of these tools to ImageJ_{man}. Statistical analysis was performed using the R software version 3.4.4 using BlandAltmanLeh [14] and irr [15] packages. The Bland–Altman plots [16,17] were produced to assess the level of agreement (width of the 95% limits of agreement [LoA]) of the cell counts in four pairwise comparisons: ImageJ_{man} – QC, ImageJ_{man} – ImageJ_{auto}, ImageJ_{man} – CellProfiler and ImageJ_{man} – Cell C. The agreement between two techniques can be judged based on the mean difference and LoA shown in the Bland–Altman plot. The mean difference closest to 0 shows that there is strong agreement between the two techniques used. The smaller the range between the upper and lower limits of the LoA demonstrates that two techniques compared can be used interchangeably. We also computed the intraclass correlation coefficients (ICC). In addition, the ability of these tools to count cells accurately at different cell densities was also determined. In comparing this, the image outputs generated from the analysis by QC, ImageJ_{auto}, CellProfiler and CellC were validated for true-positive, false-positive and false-negative. The performance of QC, ImageJ_{auto}, CellC and CellProfiler in identifying and counting cells was determined by the precision and recall of the tools based on the definitions and equations as described by others [18] (Supplementary Table 1). In order to identify true-positive, false-positive and false-negative, each cell-counting tool will produce image outputs with labelled identified cell nuclei location. The image outputs were then matched and compared with the ImageJ_{man} labelled images. We ►

Figure 2. Image processing and optimization steps. (A) The inherent image processing steps involved in the ImageJ_{auto}, QC, CellC and CellProfiler for counting cells. (B) The optimization steps involved in the ImageJ_{auto}, QC, CellC and CellProfiler before analysis start. QC: QuickCount®.

► used Photoshop for the cell nuclei identification matching as it allows us to overlay two images (image from ImageJ_{man} and image from tested software) together for quick verification. An example of verified image with all the labelled true-positive, false-positive and false-negative is shown in Figure S2.

Here, precision refers to the ability of the cell-counting tool to correctly identify cell nuclei in the images, and recall or sensitivity refers to the ability of the cell-counting tool to identify all possible cell nuclei in the images correctly. A cell-counting tool with high precision does not necessarily have high recall. To determine the accuracy of each tool, the F_1 score, which is the weighted harmonic mean of precision and recall, taking false-positives and false-negatives into account would be calculated. The F_1 score reaches its best value at 1 (with perfect precision and recall) and at its worst at 0. The formula to calculate the F_1 score is:

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

RESULTS & DISCUSSION

Performance of QC in detecting & counting cells

The user-operating workflow for QC is explained in Figure 3. Briefly, a user first selects the folder containing the images for cell counting (Figure 3A) and the list of image files will be depicted as shown in Figure 3B. Notably, several parameters can be adjusted (Figure 3C) to set ideal parameters for cell detection and the effect of these adjustments can be viewed in the real-time display (Figure 3D) with the number of counted cells indicated (Figure 3E). The user starts the analysis by clicking ‘Start Analysis’ and optimized settings can be saved as a preset for future use (Figure 3F & 3G). Cell counts are tabulated in a separate window (Figure 3H) and can be exported to comma-separated value (CSV) format. User instruc-

tions can be obtained from the manual in the Help menu (Figure 3I).

Versatility of QC in producing cell counts for different cell types

To determine the versatility of QC in counting different cell types that have different patterns of clustering during growth, we used a variety of cell lines from different cell lineages and subjected the cells to the Click-iT™ EdU cell proliferation assay. Images from these experiments were used and an average of five random views were captured for Alexa 647 and DAPI staining from each experiment, and the number of proliferating cells was determined by QC, ImageJ_{auto}, CellC and CellProfiler. The Bland–Altman analysis showed that QC comparison with ImageJ_{man} has the smallest magnitude of mean and standard deviation of difference (-3.3 ± 4.5) followed by ImageJ_{auto} (-5.1 ± 4.9), CellProfiler (-9.4 ± 21.8) and CellC (-24.8 ± 31.2). The Bland–Altman plots (Figure 4A–D) showed that the width of the 95% LoA is narrowest (17.6) for the QC comparison, followed by ImageJ_{auto} (19.1), CellProfiler (85.4) and CellC (122.4) when compared with ImageJ_{man}. CellC was found to have the poorest agreement with ImageJ_{man} and one exceptionally large difference (approximately -260.9) that belongs to the HCT116 cell line was excluded in this plot. This revision of the number of observations did not affect the overall results. Besides, this cell line did not produce outliers in the other three comparisons. The Bland–Altman plot for the CellProfiler comparison showed a ‘fanning out’ effect of the differences as the average increased. The variability of the differences increased with the size of the measurement. For the CellC comparison, it can be observed that CellC produced smaller counts when ImageJ_{man} counts were big, and vice versa. Clearly, the plots showed that all four programs tend to overestimate because the biases appear to be negative. Generally, based on Figure 4A–D, the comparisons showed that the subjects that belonged to the HCT116 and U2OS cell lines have poorer agreement with ImageJ_{man}. A comparison of the different cell lines by all five counting methods suggests that the lack of congruency in HCT116 and U2OS could be due to the clumpy nature of these cells, making it harder to delineate single cells for counting (Figure 4E). We further show that ►

Figure 3. Graphical user interface of QuickCount®. The graphical user interface has several important features: (A) a directory where folders of sample images are stored; (B) a list of all image files detected within a selected folder; (C) several parameters (brightness and contrast enhancement ratio, threshold value, Gaussian kernel size and morphology opening size) can be adjusted for optimum nuclei detection prior to applying the parameters across sample images. (D) The accuracy of detecting cell nuclei with the selected parameters can be evaluated in real time on the image. (E) The number of nuclei detected in the displayed image is depicted. (F) Settings can be saved as presets for future use. To analyze all the uploaded images, the user clicks on 'Start Analysis' (G) and data counts are tabulated automatically and can be exported to CSV format (H). The instructions are available in the user manual (I).

► the ICC for QC and ImageJ_{auto} comparisons with ImageJ_{man} were typically greater than 0.90, showing high similarity between these methods with ImageJ_{man}. The ICC appeared to be moderate for CellProfiler (0.53) and poor for CellC (0.10) (Table 1). The p-values in Table 1 show that the estimates for the ICC are significantly greater than 0 for the QC, ImageJ_{auto} and CellProfiler comparisons. However, the ICC is not significantly different from 0 for CellC comparison (p-value = 0.25). The bias for ImageJ_{man} and CellProfiler is mainly due to the discrepancy between the methods for the cell lines HT29, HCT116 and MCF7 whilst for the ImageJ_{man} – CellC comparison, HCT116, MCF7 and U2OS accounted for the low ICC value (Figure 4C & D). Overall, QC and ImageJ_{auto} showed much stronger agreement with ImageJ_{man} compared with CellProfiler and CellC.

Precision & recall of QC in cell identification & quantification

One of the issues that can compromise accuracy in automated cell counting is the

inability of a software to precisely identify and delineate single cells, particularly at high cell densities. The precision, recall and F₁ Score of QC, ImageJ_{auto}, CellC and CellProfiler in detecting and quantifying the number of proliferating cells were determined and tabulated in Table 1 and Supplementary Table S2. While all tested tools achieved over 90% precision at all densities, the recall of some tested software dropped as the cell density increased. In our analysis, only QC was able to maintain over 90% recall (with mean recall ± standard deviation: 0.955 ± 0.043; 0.929 ± 0.056; 0.921 ± 0.055 at low, medium and high densities, Table S2; Figure S3). As the cell density increased, ImageJ, CellPro-

filer and CellC produced much higher false-positive and false-negatives, thus leading to lower recall. QC has lower precision value compared with ImageJ_{auto}, CellProfiler and CellC (with mean precision ± standard deviation: 0.93 ± 0.061, 0.949 ± 0.055, 0.986 ± 0.032, 0.974 ± 0.0038, respectively at low cell density) as it had more false-positives than other tested tools due to its sensitivity in detecting cell nuclei and picking up cells that are not readily seen by the human eye. False-negatives were usually due to the inability of the programs to delineate one cell from another when there is cell clumping or when cells were grown in high densities (Figure S2). Overall, QC had higher F₁ scores compared with ImageJ_{auto}, CellProfiler and CellC at all densities (Table 2). As optimum settings were used in each of the methods as described above, the outcomes observed here are due solely to the inherent algorithms of the software in identifying and counting cells.

QC analyzes large-scale cell images in a time-efficient manner

We compared the steps taken for analysis to be performed using each tool evaluated ►

pE-300^{ultra}

Award Winning
Microscopy
Illumination

Cost effective Illumination for High
Speed Imaging.
Speed without the Cost

For more information on how CoolLED products can help you, contact us now:
t: +44 (0)1264 323040 (Worldwide)
1-800-877-0128 (USA/Canada)
w: www.CoolLED.com
e: info@CoolLED.com

CoolLED
Simply Better Control

www.CoolLED.com

Oct 18

Figure 4. Comparison of cell counts between QuickCount® and ImageJ_{man} across different types of cells. The cell lines, namely NIH3T3, 4T1, A431, A549, HT29, HCT116, MCF7 and U2OS, were seeded and stained by Click-iT™ EdU cell proliferation assay and up to six fields of images were captured from each cell line. Bland–Altman plots for assessing agreement of (A) QC, (B) ImageJ_{auto}, (C) CellProfiler (D) CellC with ImageJ_{man}. 95% limits of agreement: (A) –12.1, 5.5; (B) –14.6, 4.5; (C) –52.1, 33.3; (D) –86.0, 36.4. (E) QC and ImageJ_{auto} more accurately counts both single cells (NIH3T3) and cells that have a tendency to grow in clusters (HCT116 and U2OS) as compared to CellC and CellProfiler. QC: QuickCount®.

► in this study (Figure 2B). For QC, parameters are easily adjustable using the sliders, and the cell count output is displayed on the preview window in real-time. For ImageJ_{auto}, automation requires the identification of optimal parameters for each function used and processing steps were first recorded using ‘Macro Recorder’. This could be challenging for users who have little or no knowledge on macro programming [6,10,11]. For CellC, no real-time preview window is available, therefore optimal parameters have to be determined by running multiple iterations on the same image and the output image is checked for optimum parameters selection before executing batch processing. Figure S4 shows a list of output images generated by CellC in order to select the most optimal parameters for one image. In the case for CellProfiler, users first determine and insert processing modules into the pipeline and these are checked by executing the pipeline in preview mode. Here, ALEXA and DAPI images are analyzed separately and the output counts have to be tabulated for manual analysis by the user.

We compared the efficiency of QC in analyzing a large number of cell images with that of ImageJ_{man}, ImageJ_{auto}, CellC and CellProfiler. We demonstrated, in Table 2, a clear distinction in the average time taken to quantify cells per image across the five counting methods with QC having the fastest analysis output in the order of milliseconds per image while ImageJ_{auto} required an average of 1 s of analysis time per image based on the macro used. The time taken here does not include parameters adjustment or input image selection. Conversely, ImageJ_{man} required a varying amount of time from less than 30 s ►

Table 1. ICCs with 95% confidence intervals for the comparison of each program with ImageJ _{man} .			
Program	Intraclass correlation coefficient	95% confidence interval	Significance (p-value)
QuickCount®	0.95	0.91, 0.97	<0.001
ImageJ _{auto}	0.92	0.86, 0.96	<0.001
CellProfiler	0.53	0.28, 0.72	<0.001
CellC	0.10	0, 0.39	0.25

Table 2. Summary F ₁ Score for QuickCount®, ImageJ _{auto} , CellProfiler and CellC at different densities.				
Cell densities	QuickCount®	ImageJ _{auto}	CellProfiler	CellC
Low	0.9442	0.9247	0.9340	0.8910
Medium	0.9222	0.9107	0.8976	0.8884
High	0.9411	0.9002	0.9245	0.9290
Average	0.9360	0.9125	0.9188	0.9032

Table 3. Alexa 647 and DAPI stained images were analyzed by ImageJ _{man} , ImageJ _{auto} , CellC, CellProfiler and QuickCount®.			
Counting methods	Total time taken to count cells for 62 images (seconds)	Average time taken to count cells for each image (seconds)	Automation for high-throughput image analysis
ImageJ _{man}	N/A	30–90 [†]	No
ImageJ _{auto}	~62	~1	Yes [‡]
CellC	~305	~5	Yes
CellProfiler	~160	~2.5	Yes
QuickCount®	~15	~0.23	Yes

[†]Does not involve pre-processing steps, dependent on the number of cells contained in the image.
[‡]Involves macro programming to achieve high-throughput analysis.
The average time taken to count cell nuclei per image was determined across different counting methods.

► up to more than 90 s in order to analyze an image due to the manual intervention required to count the cells. CellC and CellProfiler required an average of approximately 5 s and 2.5 s, respectively to analyze each image due to several CPU intensive processing pipelines such as segmentation and fill holes. Notably, a unique feature of QC is that a folder of multiple images (up to hundreds) can be preloaded and analyzed automatically without the need for writing macros or plug-ins. Not needing to open each and every file independently means that the hands-on time is minimal. Whilst most of these tools do not come with automatic tabulation of results, in QC counts from the ALEXA and DAPI channels are tabulated independently and the percentage of proliferating cells are automatically generated. The scalability of

QC for high-throughput cell image analyses offers a great advantage in time-saving, especially for nonprogrammers.

One of the critical points in using QC is that the dual-stained paired cell images have to be labeled systematically according to the standard format, for example: 'treatment concentration (in μm) fluorescent dye (Alexa 647 or DAPI) exposure time (in ms) number of view captured (n)' so that these data can be paired for automated tabulation of the data. Failing to adhere to this labeling format will compromise the ability of QC to recognize image files to produce the cell counts. As with the use of any image analysis software, another caveat in using QC for cell counting is the quality of cell images. An image that is blurred due to impurities on the surface of coverslip could lead to inconsistency in

the average intensity of pixel across the image and inaccuracy in cell detection and counting. Notwithstanding, QC can readily be adapted for other biological processes that can be monitored by fluorescent-labelled proteins. For example, when determining the effects of promising drugs on cell cycle by using the fluorescence ubiquitination-based cell cycle indicator (FUCCI) method [19] and when quantitating cell death by using the terminal deoxynucleotidyl transferase dUTP nick end labelling (TUNEL) assay in cancer, and other areas of research [20,21].

CONCLUSION

We have developed QC, a user-friendly and scalable tool for automatic cell counting that has high precision and recall. QC can be downloaded from <http://quickcount.> ►

► cancerresearch.my/, and we anticipate that it will benefit other researchers seeking a simple, accurate and efficient cell-counting tool to assist in large-scale cell quantification.

SUPPLEMENTARY DATA

To view the supplementary data that accompany this paper please visit the journal website at: www.future-science.com/doi/suppl/10.2144/btn-2018-0072.

AUTHOR CONTRIBUTIONS

SCC and KHT designed the experiments. MZHF wrote the algorithm for QuickCount®. KHT, JKC and PSY performed the experiments and analyzed the data. JKC and DP conducted the statistical analyses. JKC and CSL designed and developed the Graphical User Interface for QuickCount®. KLB contributed essential technical expertise in image analysis methods. SCC, KHT, JKC and DP wrote the manuscript. MHZF, PSY, CSL, ZAAR and KLB reviewed and edited the manuscript.

ACKNOWLEDGMENTS

The authors thank Dr Tsung Fei Khang for advice on statistical analyses.

FINANCIAL & COMPETING INTERESTS DISCLOSURE

This study is funded by the High Impact Research, Ministry of Higher Education (HIR-MOHE) from University of Malaya (UM.C/625/1/HIR/MOHE/DENT-03) and

sponsors of Cancer Research Malaysia. Cancer Research Malaysia is a non-profit research organization. We are committed to an understanding of cancer prevention, diagnosis and treatment through a fundamental research program. The authors have no other relevant affiliations or financial involvement with any organization or entity with a financial interest in or financial conflict with the subject matter or materials discussed in the manuscript apart from those disclosed.

No writing assistance was utilized in the production of this manuscript.

OPEN ACCESS

This work is licensed under the Attribution-NonCommercial-NoDerivatives 4.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>

REFERENCES

1. Hanahan D, Weinberg RA. Hallmarks of cancer: the next generation. *Cell* 144(5), 646–674 (2011).
2. Salic A, Mitchison TJ. A chemical method for fast and sensitive detection of DNA synthesis *in vivo*. *Proc. Natl Acad. Sci. USA* 105(7), 2415–2420 (2008).
3. Graham M. *Principles of Automated Blood Cell Counters*. David B, John C (Eds). John Wiley and Sons Ltd, UK (2005).
4. Kachouie N, Kang L, Khademhosseini A. Arraycount, an algorithm for automatic cell counting in microwell arrays. *Biotechniques* 47(3), x–xvi (2009).
5. Byun J, Verardo MR, Sumengen B, Lewis GP, Manjunath BS, Fisher SK. Automated tool for the detection of cell nuclei in digital microscopic images: application to retinal images. *Mol. Vis.* 12, 949–960 (2006).
6. Choudhry P. High-throughput method for automated colony and cell counting by digital image analysis based on edge detection. *PLoS One* 11(2), e0148469 (2016).
7. Improvisation IPVCL. *Volocity User Guide*. Coventry, UK (2001–2007).

8. Molecular Devices Corporation. *The Metamorph system: Integrated system for Bioimaging*. PN, USA (2005).
9. Fadlullah MZ, Chiang IK, Dionne KR *et al*. Genetically-defined novel oral squamous cell carcinoma cell lines for the development of molecular therapies. *Oncotarget* 7(19), 27802–27818 (2016).
10. Abramoff MD, Magalhães PJ, Ram SJ. Image processing with ImageJ. *Biophotonics International* 11(7), 36–42 (2004).
11. Schneider CA, Rasband WS, Eliceiri KW. NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods* 9(7), 671–675 (2012).
12. Selinummi J, Seppala J, Yli-Harja O, Puhakka JA. Software for quantification of labeled bacteria from digital microscope images by automated image analysis. *Biotechniques* 39(6), 859–863 (2005).
13. Carpenter AE, Jones TR, Lamprecht MR *et al*. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol.* 7(10), R100 (2006).
14. Lehmert B. BlandAltmanLeh: plots (slightly extended) Bland–Altman plots. R package version 0.3.1. (2015).
15. Gamer M, Lemon J, Singh IFP. irr: various coefficients of interrater reliability and agreement. R package version 0.84. (2012).
16. Altman DG, Bland JM. Measurement in Medicine: the Analysis of Method Comparison Studies. *The Statistician* 32, 307–317 (1983).
17. Bland JM, Altman DG. Statistical methods for assessing agreement between two methods of clinical measurement. *Lancet* 1(8476), 307–310 (1986).
18. Powers DM. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. (2011).
19. Sakaue-Sawano A, Kobayashi T, Ohtawa K, Miyawaki A. Drug-induced cell cycle modulation leading to cell-cycle arrest, nuclear mis-segregation, or endoreplication. *BMC Cell Biol.* 12, 2 (2011).
20. Edison N, Reingewertz TH, Gottfried Y *et al*. Peptides mimicking the unique ARTS-XIAP binding site promote apoptotic cell death in cultured cancer cells. *Clin. Cancer Res.* 18(9), 2569–2578 (2012).
- Shen F, Su H, Fan Y *et al*. Adeno-associated viral-vector-mediated hypoxia-inducible vascular endothelial growth factor gene expression attenuates ischemic brain injury after focal cerebral ischemia in mice. *Stroke* 37(10), 2601–2606 (2006).

First draft submitted: 21 May 2018; Accepted for publication: 4 September 2018