Bioinformatics, 35(4), 2019, 706–708
doi: 10.1093/bioinformatics/bty676
Advance Access Publication Date: 1 August 2018
Applications Note

OXFORD

Bioimage informatics

TIMING 2.0: high-throughput single-cell profiling of dynamic cell–cell interactions by time-lapse imaging microscopy in nanowell grids

Hengyang Lu[1], Jiabing Li[1], Melisa A. Martinez-Paniagua[2], Irfan N. Bandey[2], Amit Amritkar[4], Harjeet Singh[3], David Mayerich[1], Navin Varadarajan[2,*] and Badrinath Roysam[1,*]

[1]Department of Electrical and Computer Engineering and [2]Department of Chemical and Biomolecular Engineering, University of Houston, Houston, TX 77204, USA, [3]Division of Pediatrics, University of Texas MD Anderson Cancer Center, Houston, TX 77030, USA and [4]Center for Advanced Computing and Data Science and Department of Mechanical Engineering, University of Houston, Houston, TX 77204, USA

*To whom correspondence should be addressed.

Associate Editor: Robert F. Murphy

Received on May 17, 2018; revised on July 12, 2018; editorial decision on July 25, 2018; accepted on July 31, 2018

Abstract

Motivation: Automated profiling of cell–cell interactions from high-throughput time-lapse imaging microscopy data of cells in nanowell grids (TIMING) has led to fundamental insights into cell–cell interactions in immunotherapy. This application note aims to enable widespread adoption of TIMING by (i) enabling the computations to occur on a desktop computer with a graphical processing unit instead of a server; (ii) enabling image acquisition and analysis to occur in the laboratory avoiding network data transfers to/from a server and (iii) providing a comprehensive graphical user interface.

Results: On a desktop computer, TIMING 2.0 takes 5 s/block/image frame, four times faster than our previous method on the same computer, and twice as fast as our previous method (TIMING) running on a Dell PowerEdge server. The cell segmentation accuracy (f-number = 0.993) is superior to our previous method (f-number = 0.821). A graphical user interface provides the ability to inspect the video analysis results, make corrective edits efficiently (one-click editing of an entire nanowell video sequence in 5–10 s) and display a summary of the cell killing efficacy measurements.

Availability and implementation: Open source Python software (GPL v3 license), instruction manual, sample data and sample results are included with the Supplement (https://github.com/RoysamLab/TIMING2).

Contact: nvaradar@central.uh.edu or broysam@central.uh.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Nanowell grids are a broadly applicable, and a very practical methodology for recording dynamic cell–cell interactions *in vitro* since they: permit continuous tracking of the same cell groups without the common disruptions, e.g. cells that exit or enter the field of view of the microscope (Merouane *et al.*, 2015) (Supplementary Fig. S1). They provide the massive throughput needed for extensive statistical sampling. Importantly, when the image analysis reveals cells of special interest, their coordinates are known precisely enough to enable robotic retrieval for downstream processing like clonal expansion or PCR. In the context of cancer immunotherapy, this method allows monitoring of effector-mediated cytotoxicity against desired target cells without the need for target cell engineering. This provides important advantages, such as the ability to use autologous or matched/primary tumor cells as target cells.

To date, given our interests, these methodologies have been applied to investigations of dynamic interactions between immune cells (e.g. T cells and NK cells) and target cells (e.g. leukemia and melanoma cancer cells) (Supplementary Table ST1). However, the method is broadly applicable to high-throughput quantitative studies of dynamic cell–cell interactions in general immunology, cellular biomanufacturing, cancer biology, differentiation, stem cell engineering, screening drugs based on interactions between different cell types within blood (Supplementary Table ST2).

Although advances in microfabrication have made the use of nanowell grids routine for most laboratories, the ability to perform quantitative analysis of the resulting data, at scale, in a timely manner, in the laboratory has remained an obstacle. *The proposed tools are aimed at eliminating this obstacle, and enabling high-throughput single-cell analyses of dynamic cell–cell interactions for a broader range of applications in cell biology.*

2 Materials and methods

A typical time-lapse imaging microscopy data of cells in nanowell grids (TIMING) experiment generates an array of multi-channel videos of 10–200 000 nanowells sampled 2–10 min apart, representing 1–2 TB of data. The proposed software (named TIMING 2.0) performs large-scale automated video array analysis. It provides a fast ‘turnkey’ solution with an efficient user interface that generates quantitative analyses of cell interaction behaviors at single-cell resolution with high yield, automation, speed and accuracy, with efficient visual confirmation. It builds on our prior work (Merouane *et al.*, 2015) that overcame the all-important yield barrier, i.e. the ability to segment and track a large percentage of nanowells with sufficient accuracy that manual proofreading is no longer needed. It used segmentation and tracking algorithms that exploited the nanowell spatial confinement to achieve >98% cell detection, segmentation and tracking accuracies, that are unattainable directly. The system ran on a centralized server, and required time-consuming data transfers to/from the server. TIMING 2.0 improves upon TIMING with a faster implementation that also avoids network transfers and allows all computations to occur on a desktop system with a graphical processing unit (GPU). It automatically counts the cells in each nanowell, identifies their types based on cell-type markers and provides dynamic measurements of size, location, shape and movements of cells; frequencies and durations of cell–cell contacts, and changes in fluorescent markers of cellular events. It enables linked visualization, and analysis of these measurements, with fast editing capabilities. In order to enable downstream statistical profiling and hypothesis testing, the data from our software can be exported in the form of spreadsheets. TIMING 2.0 incorporates the following key improvements:

2.1 Faster cell segmentation

The computationally intensive cell segmentation algorithms of TIMING were replaced by a fast iterative voting algorithm implemented on the GPU of desktop computers (Saadatifard *et al.*, 2018). The peak of the histogram of detected cell counts indicates the correct count for each video sequence, and when its value exceeds 80%, we use the corresponding count as the correct estimate. We identify frames with other cell counts, and re-segment them using the correct number of seeds (Supplementary Fig. S2).

2.2 Fast semi-automated proofreading

We developed a fast method for correcting segmentation and/or tracking errors in nanowell videos that arise due to errors in cell count estimation (most common cause). When an error is detected, the correct (visually assessed) cell count is entered manually, and the segmentation and tracking algorithms are rerun for just the affected nanowell. This procedure is illustrated in Supplementary Figure S3.

2.3 Graphical user interface (GUI)

The comprehensive TIMING 2.0 GUI (Supplementary Fig. S4) delineates nanowells, counts cells, identifies their types based on cell-type markers, and provides dynamic measurements of cell size, location, shape and movements, frequencies and durations of cell–cell contacts and fluorescent markers of cellular events. It allows selective visualization of nanowells based on chosen criteria (e.g. time of cell death, or time of first cell contact), efficient editing of segmentation and tracking results and profiling of cell interaction behaviors. The data can be exported to spreadsheets for further analysis.

3 Results and performance

For the example TIMING dataset in Supplementary Figure S1, CD19-specific CAR$^+$ (designated 19–28 z) human T cells were generated, as described previously (Liadi *et al.*, 2015); 19–28 z T cells as effectors, and NALM-6 (CD19$^+$) tumor cells as targets, were sequentially onto a nanowell array, and the kinetics of killing was monitored using TIMING. In order to test if the use of either animal derived serum [fetal bovine serum (FBS)] or autologous human serum (HS) had an impact on the killing of 19–28 z T cells, we ran parallel arrays containing either 10% FBS or 10% HS. As illustrated in Supplementary Figure S1(E), there was no appreciable difference in the frequency of T cell mediated killing under the conditions tested.

We measured the main drivers of software performance: (i) speed of automated image analysis; (ii) the segmentation accuracy and (iii) effectiveness of the graphical user interface, with a focus on the speed of making corrective edits to video sequences, as described below.

3.1 Analysis speed

Our original implementation on a Dell 910 PowerEdge server took ~10 s/block/frame, with each block representing a 6 × 6 array of nanowells. On a desktop system (Intel XEON E5-1630 CPU, NVIDIA GTX 1080, 64GB DDR3 RAM, 1TB 7, 200RPM disk), TIMING 2.0 takes ~5 s/block/frame, while also avoiding network data transfer times, while the original TIMING code takes ~20s/block/frame plus (variable) data transfer times.

3.2 Segmentation accuracy

The precision-recall curves in Supplementary Figure S2(H) show that the radial voting-based algorithm with the confinement constraint (blue) has an f-number of 0.993, which is more accurate than the previously published method with an f-number of 0.821.

3.3 Semi-automatic editing

In our validation experiments, 73% of nanowell videos containing one or more segmentation and tracking errors could be completely corrected using the fast semi-automated proofreading method, taking 5–8 s per nanowell video, much faster than frame-by-frame inspection and editing.

4 Documentation

TIMING 2.0 is written in Python—one of the most widely used language for scientific computing, and it runs on most computers (Windows, OS X and Linux). The installation on most computers requires only one command line using the widely used Conda environment. It is provided in open source form to allow non-commercial peer modification and extension. Complete documentation is provided in the supplementary Material and at the Github site noted in the abstract.

Acknowledgements

This work was completed in part with resources provided by the Center for Advanced Computing and Data Science. The authors thank Dr. Hien Nguyen for advice and access to the GPU equipped computer.

Funding

This work was funded by National Institutes of Health (R01CA174385, R00 LM011390), Cancer Prevention Research Institute of Texas (RP180466, RR140013), Melanoma Research Alliance Award (509800), Congressionally Directed Medical Research Programs (CA160591), and National Science Foundation (1705464).

Conflict of Interest: none declared.

References

Liadi,I. *et al.* (2015) Individual motile CD4(+) T cells can participate in efficient multikilling through conjugation to multiple tumor cells. *Cancer Immunol. Res.*, **3**, 473–482.

Merouane,A. *et al.* (2015) Automated profiling of individual cell-cell interactions from high-throughput time-lapse imaging microscopy in nanowell grids (TIMING). *Bioinformatics*, **31**, 3189–3197.

Saadatifard,L. *et al.* (2018) Robust Cell Detection for Large-Scale 3D Microscopy Using GPU-Accelerated Iterative Voting. *Front. Neuroanat.*, **12**, 28.