Usiigaci: Instance-aware cell tracking in stain-free phase contrast microscopy enabled by machine learning

Hsieh-Fu Tsai[a,b,*], Joanna Gajda[c,1], Tyler F.W. Sloan[d,1], Andrei Rares[e,1], Amy Q. Shen[a,*]

[a] *Micro/Bio/Nanofluidics Unit, Okinawa Institute of Science and Technology Graduate University, 1919-1 Tancha, Onna, Okinawa Japan 904-0495*
[b] *Research Fellow of Japan Society for the Promotion of Science*
[c] *AGH University of Science and Technology, Poland*
[d] *Quorumetrix Solutions, Canada*
[e] *ImagineA, The Netherlands*

Abstract

Stain-free, single-cell segmentation and tracking is tantamount to the holy grail of microscopic cell migration analysis. Phase contrast microscopy (PCM) images with cells at high density are notoriously difficult to segment accurately; thus, manual segmentation remains the de facto standard practice. In this work, we introduce Usiigaci, an all-in-one, semi-automated pipeline to segment, track, and visualize cell movement and morphological changes in PCM. Stain-free, instance-aware segmentation is accomplished using a mask regional convolutional neural network (Mask R-CNN). A Trackpy-based cell tracker with a graphical user interface is developed for cell tracking and data verification. The performance of Usiigaci is validated with electrotaxis of NIH/3T3 fibroblasts. Usiigaci provides highly accurate cell movement and morphological information for quantitative cell migration analysis.

Keywords: phase contrast microscopy, instance-aware segmentation, machine learning, convolutional neural network, stain-free cell tracking, single-cell migration

*Corresponding author
Email addresses: hsieh-fu.tsai@oist.jp (Hsieh-Fu Tsai), joannagajda5@gmail.com (Joanna Gajda), info@quorumetrix.com (Tyler F.W. Sloan), a.rares@ieee.org (Andrei Rares), amy.shen@oist.jp (Amy Q. Shen)
[1] contributed equally

1. Motivation and significance

Cell migration is a fundamental cell behavior that underlies various physiological processes, including development, tissue maintenance, immunity, and tissue regeneration, as well as pathological processes such as metastasis. Many *in vitro* as well as *in vivo* platforms have been developed to investigate molecular mechanisms underlying cell migration in different microenvironments with the aid of microscopy. To analyze single- or collective-cell migration, reliable segmentation of each individual cell in microscopic images is necessary in order to extract location as well as morphological information.

Among bright-field microscopy techniques, Zernike's phase contrast microscopy (PCM) is favored by biologists for its ability to translate phase differences from cellular components into amplitude differences, so as to make the cell membrane, the nucleus, and vacuoles more visible [1]. However, PCM images are notoriously difficult to segment correctly using conventional computer vision methods, due to the low contrast between cells and their background [2]. For this reason, many cell migration experiments still rely on fluorescent labeling of cells or manual tracking. Fluorescent labeling of cells requires transgenic expression of fluorescent proteins or cells tagged with fluorescent compounds, both of which can be toxic to cells and which require extensive validation of phenotypic changes. Although thresholding fluorescent images is relatively straightforward, cells that are in close proximity are often indistinguishable in threshold results. On the other hand, manual tracking of cell migration is labor-intensive and prone to operator error. Conducting high-throughput microscopy experiments is already possible thanks to methodology and instrumental advances, but current analytical techniques to interpret results quantitatively face major obstacles due to imperfect cell segmentation and tracking [3]. Moreover, cell movement is not the only parameter of interest in cell migration. For cell migration guided by environmental gradients, shear stress, surface topology, and electric field can also impact cell morphology [4–7].

Although many software packages have been developed for cell tracking, the majority of them handle only fluorescent images and require good thresholding results [8]. While some software tackles stain-free cell tracking, outlining each individual cell accurately to the cell boundary is difficult; thus, these packages are limited to positional tracking and cannot resolve adjacent or touching cells [8–12]. Migrating cells in ameboid or mesenchymal mode often have thin protruding cellular structures for locomotion, such as blebs or lamellipodia [13]. These structures exhibit very low contrast in PCM, which prevents reliable segmentation, even though they are essential for cell migration.

In recent years, advances in machine learning using convolutional neural networks (CNNs) have proven effective at solving computer vision problems [14–16]. Among them, Deepcell architecture, proposed by Van Valen *et al.*, has demonstrated that cells in close proximity can be segmented using pixel-wise classification of the background, the cell membrane, and the cell cytoplasm [15]. However, fluorescent staining of cell nuclei is still needed for optimal segmentation of these PCM images.

To address the above challenges, we introduce newly developed stain-free, instance-aware cell tracking software for PCM, called Usiigaci. Stain-free, instance-aware segmentation of phase contrast microscopy images is appealing to biologists because cells are free of labeling damage and their analysis does not suffer from false readings. Moreover, both locations and outlines of cells can be analyzed in their entirety.

2. Software description

2.1. Software overview

Usiigaci, pronounced as *ushi:gachi* by Hepburn romanization, is a Ryukyuan word that refers to tracing the outlines of objects, which is an appropriate description of the function of our software. Usiigaci has a semi-automated workflow consisting of three modules: a segmentation module, a tracking module, and a data processing module, all written in standard Python syntax (Figure 1).

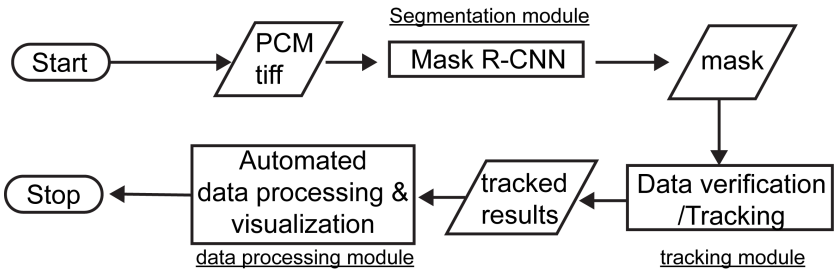

Figure 1: The all-in-one segmentation, tracking, and data processing workflow of Usiigaci.

A Mask R-CNN model pretrained with the Microsoft COCO dataset [17] was further trained using 50 manually annotated PCM images with single cell outlines as a classification class (more details in S1.4 and S1.5 in the SI document for preparing custom training data and to initiate new training). Using this trained model, PCM images are provided as input to the Mask R-CNN-based segmentation module and highly accurate instance-aware segmented masks are generated [18]. Outlines of individual cells in the images

are correctly segmented into identifiers (IDs), even if they are in close proximity. IDs are then linked and tracked in the tracking module. With the aid of a graphical user interface (GUI) in the tracking module, side-by-side comparison of PCM images and tracked masks allow users to validate segmentation and tracking results. At this point, unwanted cell tracks, such as imperfectly segmented or tracked cells, mitotic cells, or dead cells can be excluded by users prior to data processing. Thereafter, step-centric and cell-centric parameters of cell migration, as well as visualization of cell migration data are computed and generated automatically from the tracked results in the data processing module (Table S.1 in the SI document).

Based on the three modules described above, Usiigaci is an all-in-one, semi-automated solution for stain-free cell migration analysis in PCM, with a biologist-friendly workflow.

2.2. Software architecture and functionality

A diagram of segmentation and tracking modules of Usiigaci is shown in Figure 2. The segmentation module of Usiigaci is based on a Mask R-CNN model that is implemented in TensorFlow and Keras, as originally open-sourced by Matterport Inc. under the MIT license [19–21]. A detailed diagram of Mask R-CNN architecture is shown in Fig. S.3 in the SI document. The Mask R-CNN model is built upon the Faster R-CNN model that has achieved rapid identification of objects through searching regions of interest (ROIs) on feature maps [18, 22]. Raw images undergo multiple convolutional operations in a R-CNN backbone, which is composed of a residual function network (ResNet-101, [23]) and a feature pyramid network (FPN, [24]), to generate 5 feature maps (C1 to C5). ROIs are searched on feature maps using region proposal layers. An accurate instance-segmented ROI map is generated by an ROI align layer to correct for misalignment in the ROIPooling operation. After upsampling, entire outlines of individual cells are segmented into polygons bearing unique IDs in the exported mask. As a result, highly accurate, instance-aware segmentation of stain-free PCM images is realized.

After segmentation, each mask contains segmented cell outlines bearing a unique identifier (ID). The IDs are then used for linking and tracking in the tracking module built on the Trackpy library [25]. The features of an ID, such as location, equivalent diameter, perimeter, eccentricity, orientation, and true solidity, are used as parameters in Trackpy for tracking. IDs in each consecutive mask in a time-lapse experiment belonging to the same cell are searched by the Trackpy library using its default nearest neighbor search option, namely the k-dimension tree algorithm [26–28].

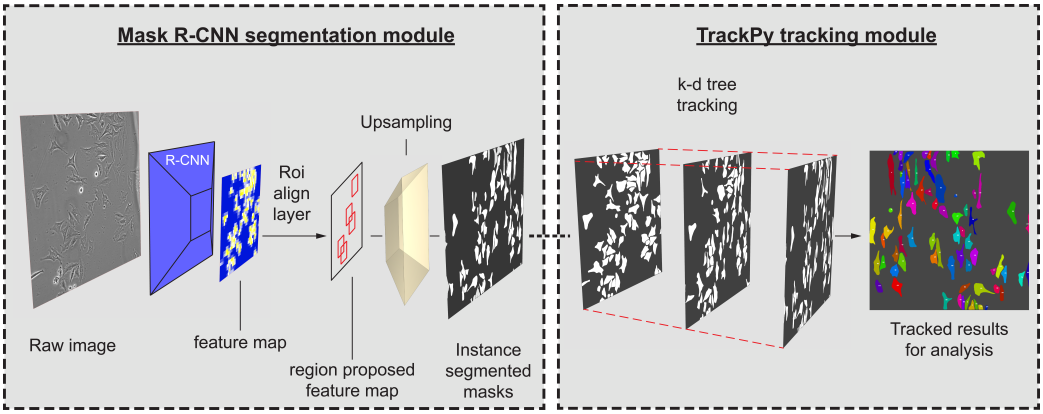

Figure 2: Diagram of segmentation and tracking modules of Usiigaci. PCM images are processed in a Mask R-CNN segmentation module with a region proposal network, which has a backbone of ResNet-101 and a feature pyramid network (FPN), to generate instance-segmented masks. Objects in the masks are linked and tracked in a Trackpy-based tracker using the k-dimensional tree algorithm. Important cell migration parameters are then computed from the tracked results.

Linking and tracking are followed by automatic post-processing, where segmentation and tracking results are corrected in two steps. In the first step, a cell wrongly segmented as two IDs is corrected by merging the two IDs. In the second step, IDs in consecutive frames belong to the same track, but suffering from interrupted events are re-linked. A GUI based on the PyQt and PyQtGraph library for the tracking module is developed so that users can verify segmentation and tracking results [29, 30]. Manual verification is important because imperfections in segmentation can cause errors in tracking. In addition, cells that undergo mitosis and cells that enter or exit the viewfield during the experiment generate tracking results that are not meaningful in single cell migration studies (Fig. S.6 in the SI document). In the GUI of the tracking module, by imposing a simple criterion, *select complete tracks*, the valid tracks IDs of which exist in every frame, can be selected. Thereafter, users can manually verify whether the tracking is correct by cross-referencing against raw images. The amount of labor in the proposed workflow is less than that associated with conventional manual tracking [4]. Subsequently, centroid and morphology parameters such as angle, perimeter, and area of each ID in valid tracks can be extracted and produced using the scikit-image library [31].

Analysis of single-cell migration data is accomplished in the data processing module to compute migration parameters for each ID throughout the time-lapse experiment (Fig. S.1.B). Several data processing libraries, including the Python data analysis library (Pandas), NumPy, and SciPy, are used

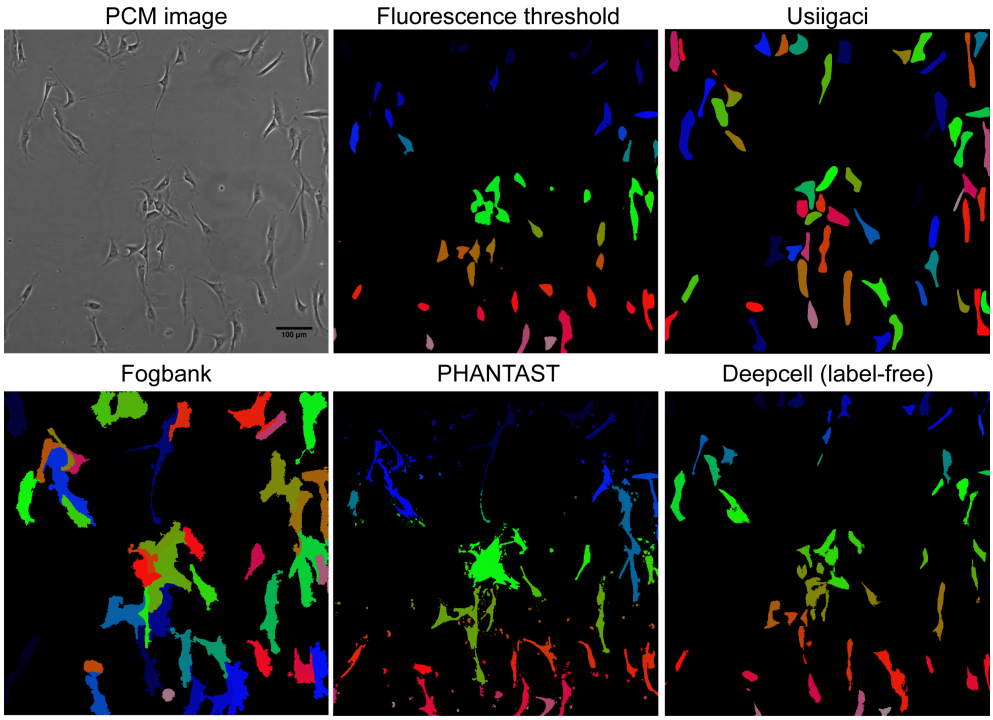

Figure 3: Microscopy of NIH/3T3 cells stained with CellTracker Green under PCM and fluorescence microscopy, compared with segmentation results of Usiigaci, Fogbank, PHANTAST, and Deepcell on the PCM image. Different color represents instances of each region of interest. In Usiigaci, Fogbank, and Deepcell, each cell is segmented into an instance outline with a unique ID and color. In segmented masks of fluorescence-thresholded or PHANTAST, cells are segmented into ROIs using the *analyze particle* function in ImageJ and filled with pseudocolors using the ROImap function in the LOCI plug-in. Usiigaci accurately segmented each individual cell with accuracy superior to that of other software.

for processing cell migration data [32–34]. Step-centric and cell-centric features, such as turning angle, net trigonometric distance, speed, orientation, and directedness are computed automatically in a Jupyter Notebook (Table S.1) [35, 36]. Moreover, automated visualization of cell migration in cell trajectory plots, box plots, and time-series plots is generated with the aid of Matplotlib and Seaborn plotting libraries (Fig. S.9) [37, 38].

3. Validation of Usiigaci

3.1. Segmentation module

Stain-free tracking of NIH/3T3 fibroblasts electrotaxis in a 300 V/m direct current electric field (dcEF) for 10 hr under PCM is used to demonstrate unique features of Usiigaci. Details of cell experiments and imaging

are described in the supplementary information. Segmentation and tracking performance of Usiigaci is benchmarked against state-of-the-art free software such as PHANTAST [11], Fogbank [12], Deepcell [15] as well as proprietary software such as Imaris and Metamorph.Segmentation results of Usiigaci and aforementioned software are shown in Figure 3 and quantitatively analyzed by segmentation evaluation metrics (Table S.2 in the SI document). Segmentation similarity can be evaluated using the mean ratio of intersection over union (mIoU), which is also known as the Jaccard index (Figure 4).

By fluorescence thresholding, thicker cell bodies can be segmented easily, but thinner structures, such as lamellipodia or blebs, often fail to be segmented and contribute to higher specificity and lower mIoU (Table 1 & Fig. S.7 in the SI document). In Fogbank and PHANTAST, images are thresholded by local contrast, thus segmentation is effective only if single cells are well isolated. The segmentation similarity achieved by Fogbank and PHANTAST is moderately high (mIoU 0.46 and 0.63), but single-cell tracking in images with high cell density is not effective using these two methods, because individual cells cannot be distinguished. By classifying cell membranes through machine learning methods, Deepcell segments high density cells better than conventional methods. However, due to the pixel-level classification methods in Deepcell, adjacent cells without clear boundaries are sometimes difficult to segment. In Usiigaci, entire outlines of cells are segmented correctly in an instance-aware fashion, even if cells are densely packed. The segmentation similarity of Usiigaci with a single trained model is 2.2 times higher than that of the fluorescence threshold method. Usiigaci's segmentation also outperforms other benchmarked segmentation software (Table 1 & Figure 4). Moreover, the segmentation speed of Usiigaci is fast in comparison to manual segmentation and benchmarked software (see Fig. S.8 in the SI document).

However, the potential limitation of Usiigaci's Mask R-CNN (essentially a machine learning method), is that segmentation accuracy may be profoundly impacted if the segmentation image is significantly different from that in the training dataset (see detailed discussion in section S2.3 in the SI document). A proper training dataset created by end users with a user-specific experimental configuration may be necessary for optimal results. The detailed description of training data preparation and training process in supplementary section S1.4 and S1.5 should help users to achieve optimal results if a new training dataset is required.

3.2. Tracking module

Mask R-CNN segments cells in an instance-aware manner such that each segmented cell possesses a unique ID (shown with pseudo-color in Figure 3).

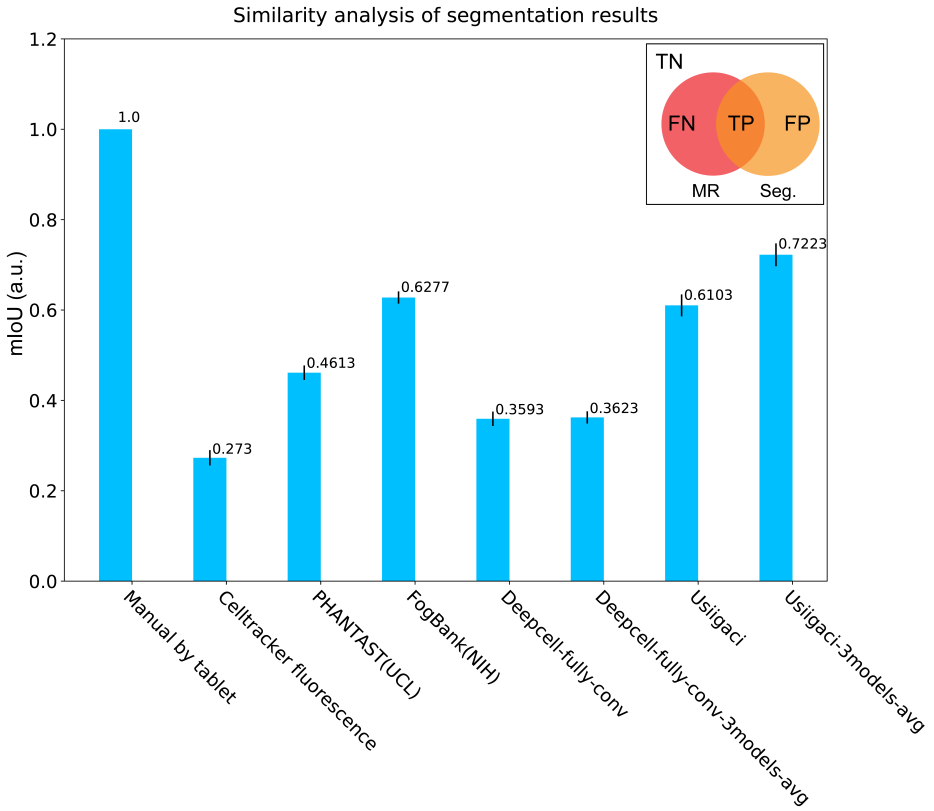

Figure 4: Segmentation similarity averaged among three NIH/3T3 cell images using various methods. MR: Manual reference; Seg.: Segmented results; FN: False negative; TP: True positive; FP: False positive; TN:True negative. Segmentation similarity is measured by the mean intersection over union between ground truth and segmented results (mIoU = $\frac{\sum_i TP_i}{\sum_i (FN_i + TP_i + FP_i)}$, as shown in the inset), or also known as the Jaccard index.

Table 1: Segmentation performance averaged among three NIH/3T3 cell images using various methods. Ch:channel

	Ch	Jaccard index	F1 score	Precision	Recall	Specificity	Accuracy
Manual	PCM	1	1	1	1	1	1
Automatic threshold	FL	0.27±0.03	0.46±0.02	0.97±0.02	0.30±0.01	1±0	0.91±0.01
PHANTAST	PCM	0.46±0.02	0.59±0.09	0.70±0.19	0.51±0.02	0.97±0.03	0.91±0.03
Fogbank	PCM	0.63±0.02	0.77±0.02	0.65±0.02	0.93±0.02	0.94±0.01	0.94±0.01
Deepcell 3models-avg	PCM	0.36±0.04	0.56±0.06	0.39±0.06	0.96±0.01	0.92±0.01	0.92±0.01
Usiigaci 3models-avg	PCM	0.72±0.01	0.85±0.01	0.83±0.02	0.87±0.01	0.95±0.04	0.96±0.01

The IDs in consecutive images are linked and tracked in the tracking module. A GUI is developed to provide manual data verification for users to identify potential errors in segmentation and tracking (Figure 5). A simple criterion,

select complete tracks, is built in the GUI for selecting tracks with IDs that exist in every frame. Imposing the criterion ensures high probability of valid tracks (Fig. S.6). Furthermore, the validity of cell tracks can be verified by users. Tracks that are biologically invalid, such as those having cells that have undergone mitosis or cell death, can be excluded manually. Usiigaci provides a labor-saving workflow while preserving the capacity for human intervention, which is essential to ensure data validity in single-cell migration analysis [39].

We characterized tracking performance using multiple object tracking (MOT) metrics and tracking quality measures on a triplicate 10-hr NIH/3T3 electrotaxis dataset (Table S.3). MOT metrics measure the performance of trackers based on how accurately the objects in every frame are tracked. Tracking quality can be understood more intuitively by classifying individual cell tracks in tracking quality measures. Detailed definition of tracking performance is discussed in the supplementary section S1.7.

The MOT performance of Usiigaci with or without manual verification is benchmarked against manual tracking as shown in Table 2 [40, 41]. In manual tracking, the multiple object tracking precision (MOTP) and multiple object tracking accuracy (MOTA) are arbitrarily defined as 1. A total of 4520 events are identified, summed from all frames. After tracking by the Usiigaci tracker, 4470 events are identified with MOTA of 91.9%. By imposing the *select complete track* criterion, events belonging to invalid tracks (Fig. S.6 B-H) are easily removed. The MOTPs describing the total error in positions of matched object-hypothesis pairs in Usiigaci before and after manual verification are 70.2% and 75.6%, which are similar to the Jaccard index in segmentation [40]. The masks of tracked cells correlate well with those by manual segmentation at pixel level, which suggests that cell movements and morphology changes can be tracked and analyzed quantitatively.

Tracking quality using the Usiigaci tracker can be understood more intuitively by classifying individual cell tracks. By manual tracking, 104 valid tracks are found among 155 total tracks. Using the Usiigaci tracker, 291 tracks are generated and many of which are erroneous due to different types of error (Fig S.6). The valid track ratio in Usiigaci is only 19.5% without manual verification. However, by the *select complete tracks* criterion, users can select only the tracks with the same ID in every frame. Valid cell tracks will be among those selected with the criterion. Users can also verify whether there are any erroneous tracks and exclude them if necessary. Five mitosis tracks exist in the remaining results and they are excluded manually. The valid tracks obtained from Usiigaci after manual verification correspond to 54% of valid tracks identified by a human operator. However, more viewfields can be analyzed to increase the number of valid tracks with the labor-saving

workflow of Usiigaci.

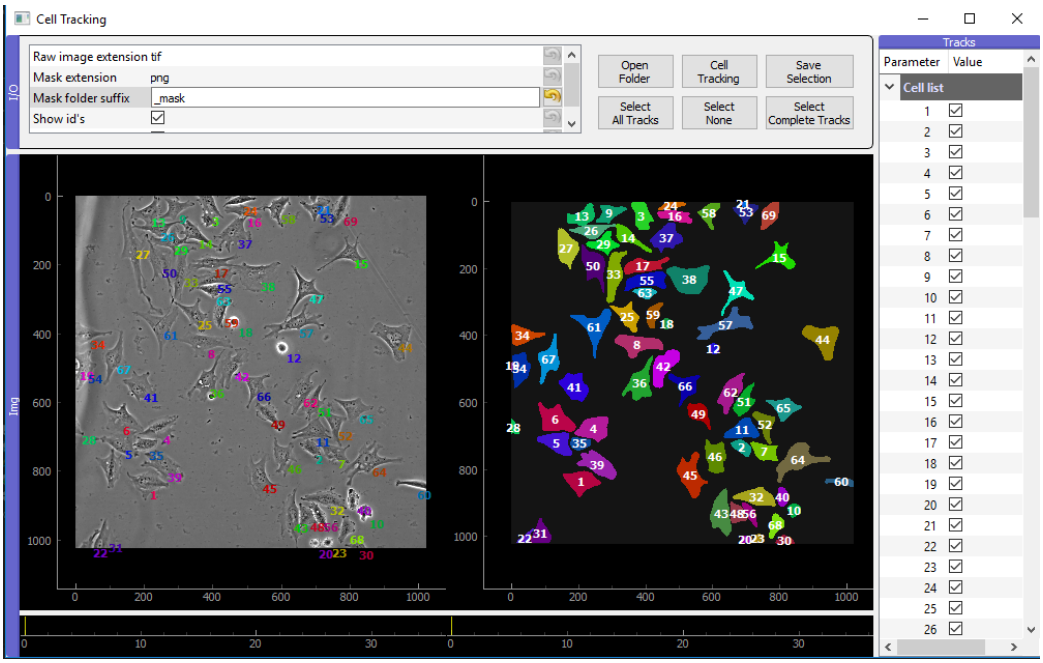

Figure 5: The GUI of the tracking module in Usiigaci. PCM images of a time-lapse experiment are shown in the left panel to compare with the Mask R-CNN segmented masks in the right panel. After tracking, cell tracks are listed on the right and users can verify data against PCM images and exclude bad cell tracks.

3.3. Data processing module

Quantitative cellular dynamics require both accurate cell segmentation and cell tracking. After tracking, the data processing module of Usiigaci generates quantitative results of step-centric and cell-centric parameters in cell migration based on the tracking results. Visualization of cell migration is carried out automatically to generate visual representations that can be understood intuitively (Fig. S.9 in the SI document).

We further examine overall accuracy in the context of cell migration among the results segmented and tracked using various methods. Directedness is a metric to show directional cell migration. Directness is defined as the average cosine between the net trigonometric distance and electric current vector (Fig. S.1B). A group of cells migrating toward the cathode has a directedness of 1, and random migrating cells possess a directedness of 0 (Table S.1). The directedness of NIH/3T3 cells in dcEF is used to benchmark the accuracy of results tracked by various tracking methods including manual tracking in ImageJ, the track object module in Metamorph, Imaris Track,

Table 2: Summary of multiple object tracking of NIH/3T3 electrotaxis after 10-hr under 300 V/m dcEF (31 frames). Metrics are compared among manual tracking and Usiigaci with and without the *select complete tracks* criterion and manual verification. MOTP: Multiple object tracking precision; MOTA: Multiple object tracking accuracy.

MOT metrics	**Manual**	**Usiigaci (unverified)**	**Usiigaci (select complete track)**
Total events	4520[a]	4470	1736
Miss events	0	145	0
Mismatch events	0	70	0
False positive events	0	165	0
MOTA	1	0.919±0.01	n/a
MOTP (mIoU)	1	0.702±0.012[b]	0.756±0.009[b]
Tracking quality measure			
Total tracks	155[c]	291[d]	61
Valid single tracks	104	56	56
Interrupted single cell tracks	0	21	0
Mitosis cell tracks	5	5	5
Entering viewfield tracks	19	19	0
Loss of tracking tracks	0	152	0
Exiting viewfield tracks	27	27	0
Mismatch tracks	0	2	0
False positive tracks	0	9	0
Valid track ratio	0.67[e]	0.19[f]	0.92[g]

[a] Total objects identified by a human operator.
[b] Mean intersection over union ratio of all matched-object pairs in mean±standard error of mean.
[c] Total cell tracks identified by a human operator.
[d] Total cell tracks generated by Usiigaci's tracker.
[e] Ratio of valid cell tracks to total cell tracks in the dataset identified by a human operator.
[f] Ratio of valid cell tracks to total cell tracks generated by Usiigaci's tracker.
[g] Ratio of valid cell tracks to total cell tracks after the *select all tracks* criterion.

and tracking with Lineage Mapper (Figure 6 & Fig. S.10). PCM images, fluorescence images, or segmented masks from either Usiigaci, PHANTAST, Fogbank, or Deepcell are used in each tracking software accordingly. Only valid cell tracks that contains cells being tracked in every frame are analyzed. Capture rate is defined as the ratio between valid cell tracks by a certain method and valid cell tracks identified manually.

While cell tracking in proprietary software such as Imaris and Metamorph yields results similar to the manual reference, both software packages only provide positional information about cells, while morphological information of cells is not available. Moreover, Imaris demands fluorescent labeling of cells to obtain good segmentation results (Table S.4).

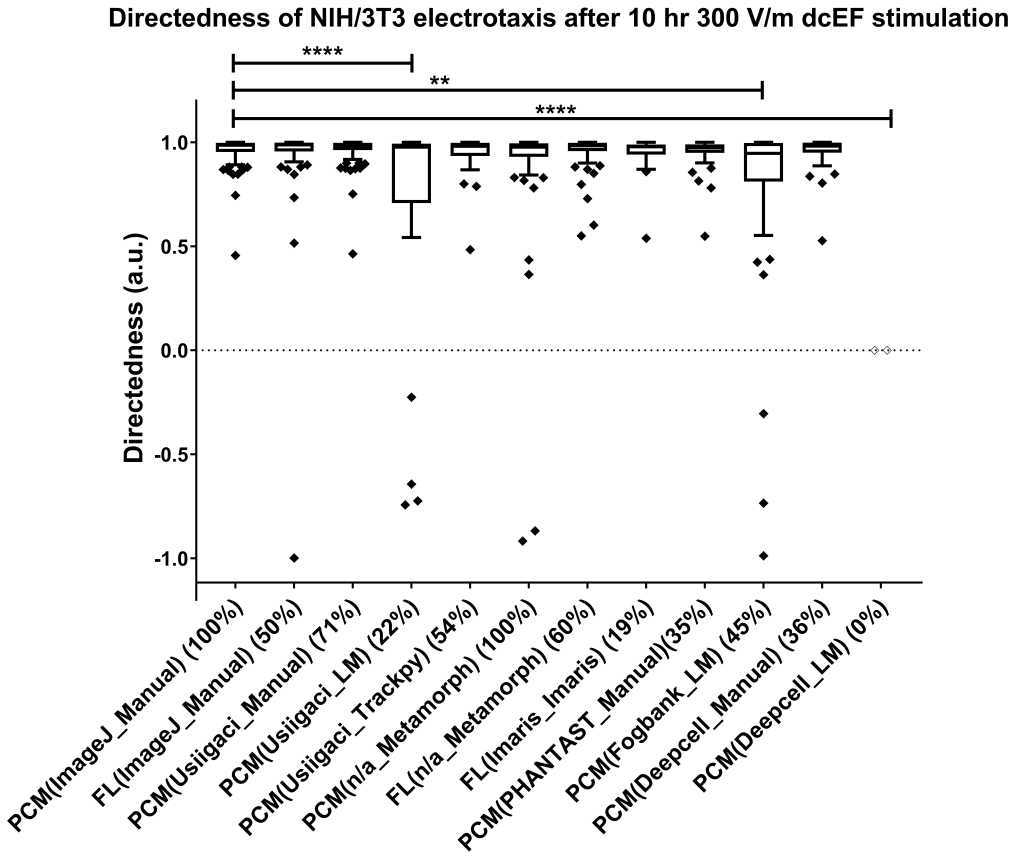

Figure 6: Directedness of NIH/3T3 electrotaxis after 10-hr, 300 V/m dcEF stimulation analyzed by different segmentation and tracking methods. Data and labels are arranged based on the type of images-(segmentation method_tracking method) (capture rate). LM:Lineage Mapper; FL:fluorescence; PCM: phase contrast microscopy; ** denotes P<0.01; **** denotes P<0.0001.

Even though open-source cell tracking software, such as Lineage Mapper is available [42], segmented data may not be directly compatible with Lineage Mapper if single cells are not segmented into individual instances correctly in every frame. Because Lineage Mapper is fully automatic, a manual verification process is not available in Lineage Mapper. Imperfect segmentation results lead to erroneous tracking results and invalid tracks cannot be excluded by users. Directedness of cells segmented by Fogbank and tracked by Lineage mapper ($P<0.01$, Tukey's post-hoc) differs from the manual reference. Cells segmented by Deepcell are not tracked well with Lineage Mapper ($P<0.0001$, Tukey's post-hoc). Therefore, segmented results from PHANTAST and Deepcell on NIH/3T3 electrotaxis cannot yield good data by tracking with Lineage Mapper. While Usiigaci's segmented masks can also be tracked using Lineage Mapper, only 22% of cells are tracked compared to the manual tracking reference. The results from Lineage Mapper are also significantly different compared to a manual reference ($P<0.0001$, one-way ANOVA), presumably due to erroneous tracking that cannot be verified manually. Misinterpretation may be made due to bad results if users do not fully grasp the inner workings of the tracking process (Figure 6).

In contrast, by segmenting and tracking with Usiigaci, 54% of cells can be automatically tracked when compared to manual tracking. Moreover, directedness and migration speed of cells analyzed by Usiigaci are comparable to the manual reference and Metamorph. Migration speed can be over- or under-estimated in Imaris or Lineage Mapper. Detailed tracking results are shown in Table S.4 in the SI document.

Usiigaci is the only automated cell tracking method that provides both cell movement and morphology change information among benchmarked software packages. With high segmentation and tracking accuracy, Usiigaci delivers quantitative cell migration analysis to biologists as an easy-to-use tool. A tutorial video of Usiigaci's usage is provided in the supplementary information (Video S.1).

4. Impact and conclusions

Usiigaci offers a reliable quantitative solution for segmentation, tracking, and analysis of cell migration in two-dimensional PCM. No label or special treatment of cells is required, so that cells can be analyzed under more natural conditions. Entire outlines of cells are automatically segmented and tracked in Usiigaci, which enables biologists to analyze both movement and morphological changes in cellular dynamics in a quantitative manner that existing software cannot provide. The labor-saving workflow also alleviates the workload in comparison to the manual cell tracking method that is conven-

tionally adopted. The manual verification function enables users to verify the tracking data and ensure data validity. The analytical capability of Usiigaci can contribute to the international effort to standardize cell migration experiments [43]. The trainable nature of the Mask R-CNN model allows Usiigaci to analyze images acquired in other bright-field microscopic techniques, and potentially for 3D cell tracking in the near future. Similar deep learning methods for biomedical image analysis are used to accomplish *in silico* labeling of cellular components instain-free images and 3D segmentation of noisy medical images [44–47]. Advances in deep learning methods for biomedical image analysis provide unique opportunities to advance biomedical discovery.

Acknowledgements

This work is supported by JSPS KAKENHI [Grant Number JP1700362]. H.-F. Tsai and A.Q. Shen also thank Okinawa Institute of Science and Technology Graduate University (OIST) for its financial support with subsidy funding from the Cabinet Office, Government of Japan. Funders had no role in study design, data collection, the decision to publish, or preparation of the manuscript. The authors acknowledge support from the Scientific Computing and Data Analysis Section, the Community Relations Section, and the Imaging Analysis Section of OIST Graduate University. The authors also thank Matterport Inc. for their Mask R-CNN implementation source code released under the MIT license for use in part of this work. The authors thank Mr. Emanuele Martini for his open-source BW_Jtrack ImageJ plugin. The authors acknowledge Ms. Tsai, Yi-Ching (lotte891@gmail.com) and Ms. Shivani Sathish from Micro/Bio/Nanofluidics Unit at OIST for assistance in preparation of illustrations in this work. The authors thank Dr. Steven Aird, OIST's technical editor for proofreading this article.

Conflict of interests

The authors declare no conflict of interests.

Supplementary Information

Supplementary information includes detailed description on cell migration experiments, microscopy protocols, annotation of training dataset, training process on the Mask R-CNN model, evaluation of multiple object tracking benchmark, and discussions on the limitation of Usiigaci. A video tutorial of Usiigaci is also attached (Video S.1).

References

[1] F. Zernike, Phase contrast, a new method for the microscopic observation of transparent objects, Physica 9 (7) (1942) 686 – 698, ISSN 0031-8914, doi:10.1016/S0031-8914(42)80035-X.

[2] N. Jaccard, N. Szita, L. D. Griffin, Segmentation of phase contrast microscopy images based on multi-scale local Basic Image Features histograms., Computer methods in biomechanics and biomedical engineering. Imaging & visualization 5 (2017) 359–367, ISSN 2168-1163, doi:10.1080/21681163.2015.1016243.

[3] R. Wollman, N. Stuurman, High throughput microscopy: from raw images to discoveries., Journal of cell science 120 (2007) 3715–3722, ISSN 0021-9533, doi:10.1242/jcs.013623.

[4] H.-F. Tsai, J.-Y. Cheng, H.-F. Chang, T. Yamamoto, A. Q. Shen, Uniform electric field generation in circular multi-well culture plates using polymeric inserts., Scientific reports 6 (2016) 26222, ISSN 2045-2322, doi:10.1038/srep26222.

[5] J. E. Moore, E. Bürki, A. Suciu, S. Zhao, M. Burnier, H. R. Brunner, J. J. Meister, A device for subjecting vascular endothelial cells to both fluid shear stress and circumferential cyclic stretch., Annals of biomedical engineering 22 (1994) 416–422, ISSN 0090-6964.

[6] R. Steward, D. Tambe, C. C. Hardin, R. Krishnan, J. J. Fredberg, Fluid shear, intercellular stress, and endothelial cell alignment., American journal of physiology. Cell physiology 308 (2015) C657–C664, ISSN 1522-1563, doi:10.1152/ajpcell.00363.2014.

[7] C. J. Bettinger, R. Langer, J. T. Borenstein, Engineering substrate topography at the micro- and nanoscale to control cell function., Angewandte Chemie (International ed. in English) 48 (2009) 5406–5415, ISSN 1521-3773, doi:10.1002/anie.200805179.

[8] E. Meijering, O. Dzyubachyk, I. Smal, Methods for cell and particle tracking, in: Methods in enzymology, vol. 504, Elsevier, 183–200, 2012.

[9] M. E. Ambühl, C. Brepsant, J.-J. Meister, A. B. Verkhovsky, I. F. Sbalzarini, High-resolution cell outline segmentation and tracking from phase-contrast microscopy images., Journal of microscopy 245 (2012) 161–170, ISSN 1365-2818, doi:10.1111/j.1365-2818.2011.03558.x.

[10] F. P. Cordelières, V. Petit, M. Kumasaka, O. Debeir, V. Letort, S. J. Gallagher, L. Larue, Automated cell tracking and analysis in phase-contrast videos (iTrack4U): development of Java software based on combined mean-shift processes, PloS one 8 (11) (2013) e81266.

[11] N. Jaccard, L. D. Griffin, A. Keser, R. J. Macown, A. Super, F. S. Veraitch, N. Szita, Automated method for the rapid and precise estimation of adherent cell culture characteristics from phase contrast microscopy images., Biotechnology and bioengineering 111 (2014) 504–517, ISSN 1097-0290, doi:10.1002/bit.25115.

[12] J. Chalfoun, M. Majurski, A. Dima, C. Stuelten, A. Peskin, M. Brady, FogBank: a single cell segmentation across multiple cell lines and image modalities, Bmc Bioinformatics 15 (1) (2014) 431.

[13] C. T. Mierke, The fundamental role of mechanical properties in the progression of cancer disease and inflammation, Reports on Progress in Physics 77 (7) (2014) 076602.

[14] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer, 234–241, 2015.

[15] D. A. Van Valen, T. Kudo, K. M. Lane, D. N. Macklin, N. T. Quach, M. M. DeFelice, I. Maayan, Y. Tanouchi, E. A. Ashley, M. W. Covert, Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments, PLoS computational biology 12 (11) (2016) e1005177.

[16] H. Niioka, S. Asatani, A. Yoshimura, H. Ohigashi, S. Tagawa, J. Miyake, Classification of C2C12 cells at differentiation by convolutional neural network of deep learning using phase contrast images., Human cell 31 (2018) 87–93, ISSN 1749-0774, doi:10.1007/s13577-017-0191-9.

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common Objects in Context, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, ISBN 978-3-319-10602-1, 740–755, 2014.

[18] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proc. IEEE Int. Conf. Computer Vision (ICCV), 2980–2988, doi: 10.1109/ICCV.2017.322, 2017.

[19] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, TensorFlow: A System for Large-scale Machine Learning, in: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, OSDI'16, USENIX Association, Berkeley, CA, USA, ISBN 978-1-931971-33-1, 265–283, URL http://dl.acm.org/citation.cfm?id=3026877.3026899, 2016.

[20] F. Chollet, et al., Keras, https://keras.io, 2015.

[21] W. Abdulla, Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow, https://github.com/matterport/Mask_RCNN, 2017.

[22] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (6) (2017) 1137–1149, ISSN 0162-8828, doi:10.1109/TPAMI.2016.2577031.

[23] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778, 2016.

[24] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature Pyramid Networks for Object Detection, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), ISSN 1063-6919, 936–944, doi:10.1109/CVPR.2017.106, 2017.

[25] D. B. Allan, T. Caswell, N. C. Keim, C. M. van der Wel, trackpy: Trackpy v0.4.1, doi:10.5281/zenodo.1226458, URL https://doi.org/10.5281/zenodo.1226458, 2018.

[26] J. L. Bentley, Multidimensional Binary Search Trees Used for Associative Searching, Commun. ACM 18 (9) (1975) 509–517, ISSN 0001-0782, doi:10.1145/361002.361007, URL http://doi.acm.org/10.1145/361002.361007.

[27] J. C. Crocker, D. G. Grier, Methods of digital video microscopy for colloidal studies, Journal of colloid and interface science 179 (1) (1996) 298–310.

[28] S. Maneewongvatana, D. M. Mount, On the efficiency of nearest neighbor searching with data clustered in lower dimensions, in: International Conference on Computational Science, Springer, 842–851, 2001.

[29] R. Computing, PyQt, PyQt is available online at http://www. riverbankcomputing. co. uk/, visited on June 13.

[30] L. Campagnola, PyQtGraph-scientific graphics and GUI library for python, 2016.

[31] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, scikit-image contributors, scikit-image: image processing in Python., PeerJ 2 (2014) e453, ISSN 2167-8359, doi:10.7717/peerj.453.

[32] W. McKinney, et al., Data structures for statistical computing in python, in: Proceedings of the 9th Python in Science Conference, vol. 445, Austin, TX, 51–56, 2010.

[33] T. E. Oliphant, A guide to NumPy, vol. 1, Trelgol Publishing USA, 2006.

[34] T. E. Oliphant, SciPy: Open source scientific tools for Python, Computing in Science and Engineering 9 (2007) 10–20.

[35] H.-F. Tsai, C.-W. Huang, H.-F. Chang, J. J. Chen, C.-H. Lee, J.-Y. Cheng, Evaluation of EGFR and RTK signaling in the electrotaxis of lung adenocarcinoma cells under direct-current electric field stimulation, PLoS One 8 (8) (2013) e73418, doi:10.1371/journal.pone.0073418.

[36] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. E. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. B. Hamrick, J. Grout, S. Corlay, et al., Jupyter Notebooks-a publishing format for reproducible computational workflows., in: ELPUB, 87–90, 2016.

[37] J. D. Hunter, Matplotlib: A 2D graphics environment, Computing in science & engineering 9 (3) (2007) 90–95.

[38] M. Waskom, O. Botvinnik, D. O'Kane, P. Hobson, S. Lukauskas, D. C. Gemperline, T. Augspurger, Y. Halchenko, J. B. Cole, J. Warmenhoven, J. de Ruiter, C. Pye, S. Hoyer, J. Vanderplas, S. Villalba, G. Kunter, E. Quintero, P. Bachant, M. Martin, K. Meyer, A. Miles, Y. Ram, T. Yarkoni, M. L. Williams, C. Evans, C. Fitzgerald, Brian, C. Fonnesbeck, A. Lee, A. Qalieh, mwaskom/seaborn: v0.8.1 (September 2017), doi:10.5281/zenodo.883859, 2017.

[39] P. Masuzzo, L. Huyck, A. Simiczyjew, C. Ampe, L. Martens, M. Van Troys, An end-to-end software solution for the analysis of high-throughput single-cell migration data., Scientific reports 7 (2017) 42383, ISSN 2045-2322, doi:10.1038/srep42383.

[40] K. Bernardin, R. Stiefelhagen, Evaluating multiple object tracking performance: the CLEAR MOT metrics, Journal on Image and Video Processing 2008 (2008) 1.

[41] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, K. Schindler, MOT16: A benchmark for multi-object tracking, arXiv preprint arXiv:1603.00831 .

[42] J. Chalfoun, M. Majurski, A. Dima, M. Halter, K. Bhadriraju, M. Brady, Lineage mapper: A versatile cell and particle tracker, Scientific Reports 6 (2016) 36984, doi:10.1038/srep36984.

[43] P. Masuzzo, L. Martens, et al., An open data ecosystem for cell migration research, Trends in cell biology 25 (2) (2015) 55–58.

[44] E. M. Christiansen, S. J. Yang, D. M. Ando, A. Javaherian, G. Skibinski, S. Lipnick, E. Mount, A. O'Neil, K. Shah, A. K. Lee, et al., In silico labeling: Predicting fluorescent labels in unlabeled images, Cell 173 (3) (2018) 792–803.

[45] J. Kimmel, A. Brack, W. Marshall, Deep convolutional and recurrent neural networks for cell motility discrimination and prediction .

[46] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, O. Ronneberger, 3D U-Net: learning dense volumetric segmentation from sparse annotation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 424–432, 2016.

[47] M. Xu, D. P. Papageorgiou, S. Z. Abidi, M. Dao, H. Zhao, G. E. Karniadakis, A deep convolutional neural network for classification of red blood cells in sickle cell anemia, PLoS computational biology 13 (10) (2017) e1005746.

Required Metadata

Current code version

Nr.	**Code metadata description**	**Please fill in this column**
C1	Current code version	v1.0
C2	Permanent link to code/repository used for this code version	$https://github.com/oist/Usiigaci$
C3	Legal Code License	MIT License
C4	Code versioning system used	git
C5	Software code languages, tools, and services used	Python, TensorFlow, Keras, Trackpy, NumPy, SciPy, Pandas, PyQtGraph
C6	Compilation requirements, operating environments & dependencies	Ubuntu 16.04 Linux, Python3.4+, CUDA9.1, TensorFlow 1.4, Keras 2.1
C7	If available Link to developer documentation/manual	None
C8	Support email for questions	hsieh-fu.tsai@oist.jp

Table 3: Code metadata