SoftwareX 13 (2021) 100656

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

TraCurate: Efficiently curating cell tracks

S. Wagner [a,1], K. Thierbach [a,c,1], T. Zerjatke [a], I. Glauche [a], I. Roeder [a,b], N. Scherf [a,c,*]

[a] Institute for Medical Informatics and Biometry, Carl Gustav Carus Faculty of Medicine, TU Dresden, Germany
[b] National Center of Tumor Diseases (NCT), Partner Site Dresden, Germany
[c] Max Planck Institute for Human Cognitive and Brain Sciences, Leipzig, Germany

ARTICLE INFO

Article history:
Received 18 August 2020
Received in revised form 3 December 2020
Accepted 4 January 2021

Keywords:
Image analysis
Single cell tracking
Lineage tracing
Tracking tool

ABSTRACT

TraCurate is an open-source software tool to curate and manually annotate cell tracking data from time-lapse microscopy. Although many studies of cellular behaviour require high-quality, long-term observations of single cells across several generations, automated tracking of individual cells is often imperfect and typically yields fragmented results that still contain many errors. TraCurate supports the user to efficiently curate and extract complete cell tracks and genealogies from a variety of cell tracking data. Source code and binary packages for TraCurate and all related tools are freely available for Linux, macOS, and Windows at https://tracurate.gitlab.io/.

Code metadata

Current Code version	*commit 0dbcd528e357333fe0c6bfe6c052d71a0dd06146 (rev977)*
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-20-00027
Legal Code License	*LGPLv3*
Code Versioning system used	*git*
Software Code Language used	*C++, QML*
Compilation requirements, Operating environments & dependencies	*Qt 5.8, HDF5 1.8.20, GNU make*
If available Link to developer documentation / manual	*Makefiles for different platforms are available at* https://gitlab.com/tracurate/tracurate/-/tree/master/aux
Support email for questions	sebastian.wagner3@tu-dresden.de

Software metadata

Current software version	*1.0*
Permanent link to executables of this version	*OS X:* https://tracurate.gitlab.io/downloads/binaries/tracurate-rev977_osxbuild.dmg *Linux (glibc 2.30+):* https://tracurate.gitlab.io/downloads/binaries/tracurate-rev977_linuxbuild.zip *Linux (glibc 2.23+):* https://tracurate.gitlab.io/downloads/binaries/tracurate-rev977_linuxoldbuild.zip *Windows:* https://tracurate.gitlab.io/downloads/binaries/tracurate-rev977_windowsbuild.exe
Legal Software License	*LGPLv3*
Computing platform / Operating System	*OS X, Linux (glibc 2.30+), Linux (glibc 2.23+), Windows*
Installation requirements & dependencies	*OS X: app-bundle with dependencies included, supported Version OS X Mojave 10.14* *Linux: shell script for running application, directory with libraries included, supported distributions: Ubuntu 16.04 and 18.04* *Windows: stand-alone static binary, unsupported, has known problems with HiDPI-scaling on Windows 10*
If available Link to user manual - if formally published include a reference to the publication in the reference list	*User Manual:* https://gitlab.com/tracurate/tracurate/blob/master/doc/manual/manual.md *Additional resources and downloads:* https://tracurate.gitlab.io/downloads/
Support email for questions	nscherf@cbs.mpg.de sebastian.wagner3@tu-dresden.de

1. Introduction

To understand how cells form and maintain tissues, organs and entire organisms, we need to systematically observe and

* Corresponding author at: Institute for Medical Informatics and Biometry, Carl Gustav Carus Faculty of Medicine, TU Dresden, Germany.
E-mail address: nscherf@cbs.mpg.de (N. Scherf).

[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.softx.2021.100656

analyse cellular dynamics. Time-lapse microscopy enables the continuous monitoring of individual cells or cell ensembles over time [1–4]. In a typical time-lapse experiment, the number of cells, time points, or repetitions easily exceed the capabilities of manual analysis. Thus, computer-assisted or automated cell tracking is required. In this work, cell tracking refers to the process in which cells are detected (i.e. their positions or shapes) and linked between successive observations. This task is not trivial, and current computational approaches are far from perfect. The required accuracy of cell tracking varies considerably between different biological questions: Migration patterns or shape dynamics can often be analysed directly from the results of automated tracking algorithms with limited accuracy. In contrast, studies of cellular development often require reliable long-term tracking and unbiased reconstruction of positional and divisional/genealogical information. Automated tracking algorithms are currently not accurate enough to meet such high standards, as the temporal alignment of cell objects is often error-prone and ambiguous [3,5]. Alternatively, completely manual tracking approaches are tedious and thus, severely limit the scale of feasible experiments. TraCurate aims to fill the gap by providing a flexible workflow with an easy to use front-end for the manual correction of erroneous or incomplete cell tracking data, initially resulting from automated approaches.

2. Problems and background

Over the past two decades, many tools for cell segmentation and tracking have been developed, such as [5–14]. Most available tools are either fully automated and thus imperfect, or completely manual and thus not applicable to large-scale analysis. A small number of approaches aim to combine the advantages of both manual and automated analysis. The popular TrackMate plugin for ImageJ/Fiji [15] is a good example that allows the user to perform automated particle tracking and to correct the results where necessary. However, TrackMate is limited to tracking the position of cells only [9], while ignoring shape information. This is a severe limitation for downstream analysis, although an interesting workaround in combinations with a cell segmentation tool [16] was recently published in [17]. A variety of semi-automated tools tries to integrate manual and automated analysis by allowing the user to correct mistakes and then automatically recompute the tracking [10]. While this is a very efficient approach, it typically needs to be optimised to a particular domain (e.g. to track specific cell types or to a particular microscopy modality). Therefore, these tools are typically not easily generalisable to other experiments. Further, they are often designed in a non-modular way as they tightly integrate automated tracking and data curation [11]. This combined approach makes it harder to exchange certain parts of the workflow (such as cell detection) and to adapt it to a specific experimental condition. As another limitation, some tools like CellProfiler [13] are not directly available on all platforms.

Against this background, there is a clear need for software solutions to connect the advantages of automated and manual cell tracking in a flexible workflow that allows users to mix and match cell detection/segmentation and tracking tools to retrieve accurate cell tracks and genealogies for their particular experimental situation. To this end, we developed TraCurate as a tool to simplify and accelerate this process for a wide range of applications. The flexibility of the workflow and the cross-platform integration make TraCurate an ideal solution to assist manual correction of tracking data for various settings. TraCurate reads automatically generated (typically imperfect) tracks (called AutoTracklets) and allows the user to manually curate them yielding complete tracks (called Tracklets) that can be further combined into genealogies (lineage trees) by adding TrackEvents. Our modular approach explicitly separates the two aspects of automated tracking and manual curation: TraCurate does not make restrictive assumptions about what is being tracked (e.g. it could be entire organisms) or what is being analysed (e.g. cell shape dynamics) and thus, integrates with a variety of experiment and analysis workflows, while not forcing the user to apply a fixed toolset for each sub-task. The integration of our software into existing workflows is a significant benefit, enabling users to choose the best tools for each part of their workflow instead of being stuck with a single, monolithic application.

3. Software framework

3.1. Software architecture

Overview

The TraCurate software was initially designed for tracking individual cells in ex vivo assays imaged with widely available light or fluorescence microscopes. TraCurate and its accompanying tools, tcConvert, IJTracker, and tcImport, provide a complete workflow to track cells from 2D time-lapse imaging, to import and manually correct the tracking and segmentation results and to export curated data sets (Fig. 1) for downstream analysis. The TraCurate interface allows the user to import tracking data and displays the original images overlaid with the tracking results (labelled cell masks). The user interface is equipped with different, specialised options for user-interaction, allowing the user to manually join disconnected tracks of the same object and to create cellular genealogies that extend over longer periods and multiple divisions. TraCurate further provides a simple interface to adapt (i.e. to segment, delete, split, or merge) the imported cell masks. It was our major goal to make TraCurate compatible with existing cell tracking workflows. Thus, we provide tools and scripts to integrate TraCurate with other software (see Fig. 1), so the user can:

- Convert available cell tracking data from existing file formats produced by a variety of established tracking software such as [5,13,18,19] to TraCurate's HDF5 file format using *tcConvert*. For a broader overview of existing tracking methods see [5,20].
- Create new tracks from cell masks using our simple, automated tracking plugin *IJTracker*. To create these masks, we recommend using specialised tools for segmentation such as [16,21–23].
- Import the curated tracking data for downstream analysis directly into R (https://gitlab.com/tracurate/tcImport) or Python (https://gitlab.com/tracurate/pytcimport) using *tcImport*.

Data Import and Export

The converter *tcConvert* (https://gitlab.com/tracurate/tcConvert) was developed as a stand-alone *Julia* [24] script. We chose Julia for its excellent performance (compared to Python or R) in the conversion process that enables fast processing even when converting large high-resolution time-lapse data. *tcConvert* supports various file formats to import cell tracking data. It directly works with HDF5 files from the trainable segmentation and tracking tool *ilastik* [24]. *tcConvert* further supports importing data from the recently introduced *biotracks* format, a community-driven format developed by the Cell Migration Standardisation Organisation (CMSO). *Biotracks* seeks to standardise and unify different cell tracking data formats and provides a tool to convert from many different formats into the *biotracks* standard (see [25] for more details and supported formats). By supporting

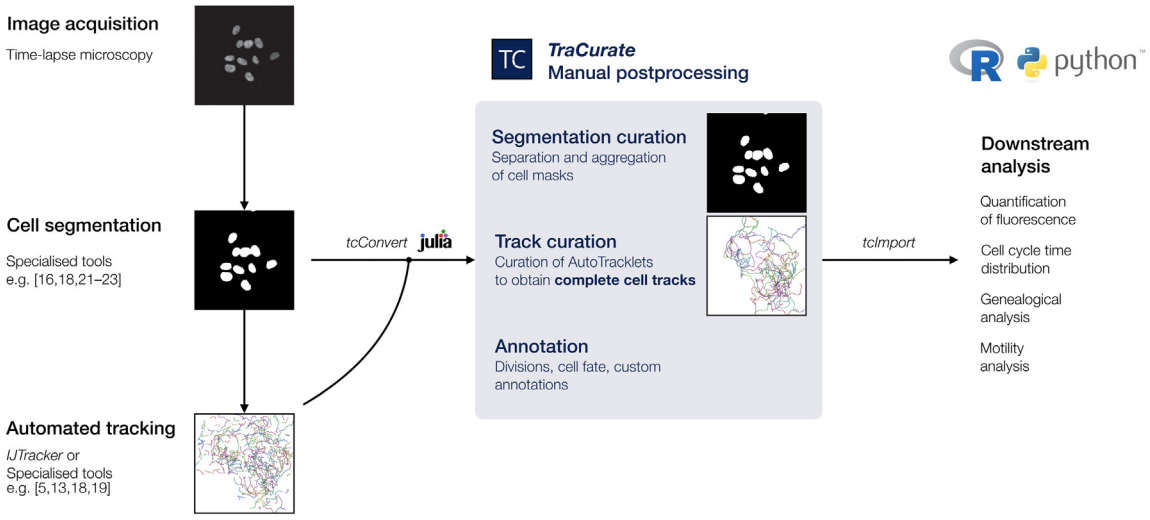

Fig. 1. General overview of a representative cell tracking workflow. Segmentation and tracking results from external tools (left) can be imported and manually corrected and annotated in TraCurate (middle). The curated results can then be exported for downstream data analysis.

the biotracks format, TraCurate works with a wide (and growing) variety of tracking tools. While the *biotracks* format can be used to standardise and annotate entire experiments (e.g. record imaging settings) we are only adopting the imaging data part, in which a JSON file is used for the metadata and a CSV files providing actual cell positions. If other, custom tracking tools are used, tcConvert provides the functionality to import basic features from a generic CSV format by directly specifying the respective columns (see documentation at https://gitlab.com/tracurate/tracurate/-/tree/master/doc/manual).

We further supply IJTracker (https://gitlab.com/tracurate/IJTracker_Plugin) as part of the TraCurate toolset to support workflows where only cell segmentation masks (or probability maps) but no tracking is available. IJTracker is a Fiji plugin that implements a simple, unbiased nearest-neighbour approach for automated generation of AutoTracklets that can then be curated. This plugin generates tracking data in an XML-based format that can be converted to the TraCurate HDF5 format.

Finally, we provide tcImport to facilitate downstream analysis and visualisation of the curated tracking results. tcImport is a package that allows the user to conveniently import TraCurate's HDF5 output into R (https://r-project.org) as well as Python (https://www.python.org), which are two of the most widely used programming languages in biomedical data analysis.

User Interface

TraCurate is implemented in a modular way to allow future extensions of its functionality. For the graphical user interface (GUI), we separated the different views (described under Software functionalities) from the data model and mechanisms for the program logic. All views are implemented along similar principles to simplify future additions to the GUI. Import and export functionality are abstracted via particular interfaces. The detailed usage of TraCurate interface and its accompanying tools is described in detail in the TraCurate User Manual, available from the website (https://tracurate.gitlab.io/). The process to compile TraCurate from source is completely automated and reproducible via GitLab's continuous integration (CI), to act as a reference system for developers that want to compile TraCurate and to ensure a reproducible build.

3.2. Software functionalities

TraCurate distinguishes two types of tracks: Tracklets and AutoTracklets. Both consist of a series of labelled cell masks across time points (frames). AutoTracklets describe the original, imported result from an external tracking algorithm, which needs to be corrected. While working with TraCurate, the user creates Tracklets from the incomplete AutoTracklets and links these Tracklets to create a full genealogy. Tracklets themselves are connected to other Tracklets via TrackEvents. The TrackEvents are used to annotate the beginning and the end of a Tracklet, while they are also used to define subsequent Tracklet(s) or the status of a Tracklet at the end of the observation period. Furthermore, TrackEvents are the central instance to annotate cell division or the case when two cells are falsely merged into a single segmentation mask. TrackEvents are also used to terminate a branch of the resulting genealogy when a cell undergoes cell death, or the experiment ended.

TraCurate's GUI incorporates different views to efficiently support the curation process: the *tracking view*, *segmentation view*, and the *project view*.

The *tracking view* (see Fig. 2) assists the user in creating Tracklets and provides contextual information about the currently selected cell and its respective (Auto-)Tracklet. Specialised interaction strategies help the user to efficiently curate and merge automatically generated tracks [26]. The user can select predefined TrackEvents (such as cell division and merging of cells), and assign them to a given Tracklet. Additional user-defined annotations (e.g. as free text or from a controlled vocabulary of biomedical ontologies [25,27–31]) can be assigned to objects and Tracklets to be available for downstream analysis.

In the *segmentation view*, the cell masks can be split and merged if needed for individual frames (note that there is no automated propagation of corrected masks). Additionally, TraCurate provides an interactive flood-fill method to add new cell masks from scratch. The flood fill algorithm is very efficient for segmenting isolated cells or even touching cells as long as there is a thin visible boundary (in terms of image brightness) between neighbouring cells. However, for very crowded situations, this process is less reliable and would require additional splitting of masks. In its current version, TraCurate only supports the manual correction of segmentation masks. If too many segmentation errors occurred during tracking, this manual correction could become infeasible. In this case, we recommend using trainable segmentation tools [16,18–21] to create more reliable masks for crowded and complex datasets.

The *project view* includes general information on the project, such as an overview of the created Tracklets or the time spent, and allows the specification of user-defined annotations.

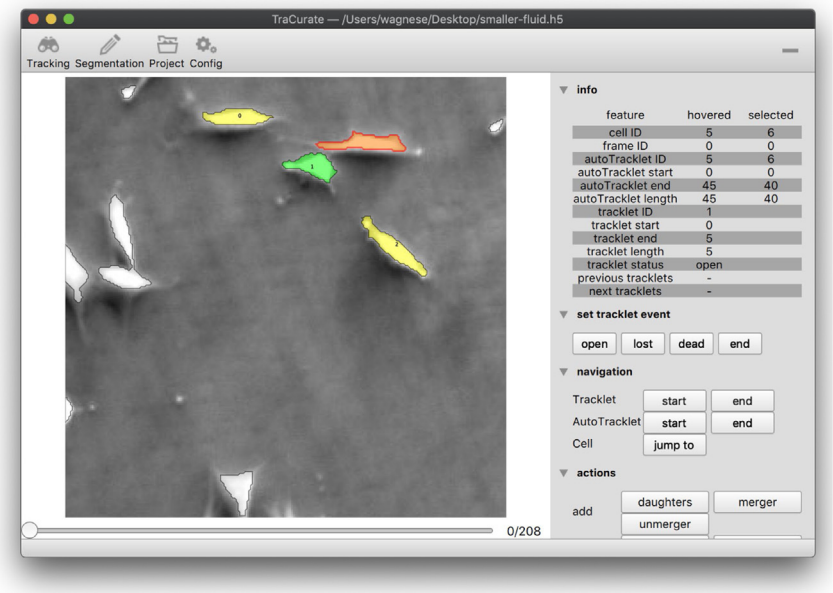

Fig. 2. Tracking view of TraCurate. The left panel shows the tracking area. The menu on the right displays information about selected objects and provides different tools for navigating, editing and annotating tracks.

4. Implementation

TraCurate is implemented in C++11 using Qt5 as a high-level library for the creation of a flexible GUI via QML and abstractions such as different data containers. We rely on the HDF5 library as the basis for our underlying data format as it allows for a compact representation of the tracking data along with additional information for a broad spectrum of potential applications. Data from HDF5 files are represented in-memory using a class hierarchy that mimics the file format itself. The image data is an exception here as it is only loaded when the image is displayed in the GUI to keep a low memory profile and enable the responsive editing of high-resolution videos. For data export, we provide the option to save either the complete HDF5 file or only selected parts (e.g. the tracking results but no image data) to minimise storage space usage. Additionally, we provide a package for both R and Python, tcImport, to load the curated Tracklets and image data into common R or Python data structures for subsequent statistical analysis.

5. Illustrative example

TraCurate (and its preceding development versions) has been used for several applications and was instrumental for many of our own studies [33–35]. We briefly illustrate a recent project, in which we studied the regulation of cell cycle progression on the single-cell level [32]. For this purpose, we imaged retinal pigmented epithelial (RPE) cells in five-minute intervals over three days (Fig. 3A). Cells carried an endogenous fluorescent marker on the DNA replication protein PCNA, which was used for segmentation and tracking of the cells. After applying our Fiji plugin IJTracker to create AutoTracklets (Fig. 3B), we used TraCurate to combine and correct these AutoTracklets in order to obtain accurate tracks over complete cell cycles (Fig. 3C). In our work, we established PCNA as an all-in-one marker that in addition to cell segmentation and tracking can also be used for classifying all cell cycle stages (Fig. 3D). Extracting fluorescence information only from the uncurated AutoTracklets, we cannot obtain a continuous quantification of the intracellular kinetics within one cell (Fig. 3E). However, extracting the same information after curation with TraCurate, we obtain continuous fluorescence time courses over the complete cell cycle (Fig. 3F), which then allow us to determine cell cycle phases based on signal intensity and intracellular distribution. For this application, TraCurate was an indispensable tool to obtain a large number of around 1000 highly accurate complete cell tracks in a semi-automated, time-efficient manner (one to two dozen complete cell tracks per hour).

In order to demonstrate TraCurate's functionality in a comprehensive example, we used a freely available 2D+Time data set from the Cell Tracking Challenge (available at http://celltrackingchallenge.net/2d-datasets/). We chose a time-lapse experiment on pancreatic stem cells from Rattus norvegicus specimens [36]. As this image sequence is also used to illustrate the workflow for segmentation and tracking using IJTracker on our website, the data set was cropped and reduced in length to decrease the file size. The supplementary video sequence (ref. Video 1) visually illustrates the process of tracking a specific cell and its progeny. Furthermore, the video demonstrates the available features for adjusting cell segmentation, such as splitting and combining outlines, deleting them and recreating a faulty segmentation with a different threshold.

6. Conclusions

We presented TraCurate, a software tool for the curation of automatically generated cell tracks and cell masks. Due to its modular nature, TraCurate integrates well into existing workflows and allows biologists to manually correct or validate the tracking when high-quality results are required. Most tools for the correction of automatically generated tracks are integrated into existing tracking software. Thus, biologists are often limited to a particular software solution that might not be optimal for their specific needs. We believe that a modular approach can increase the overall quality of cell tracking results, as each part of the tracking process can be handled by the optimal tool available for a specific situation. Thus, TraCurate can be combined with a variety of tools to segment and track cells and to analyse the data in the end. We have successfully used TraCurate and its preceding prototypes in a number of scientific projects and collaborations. Here, we provide this open-source tool for a larger biology and bioinformatics community.

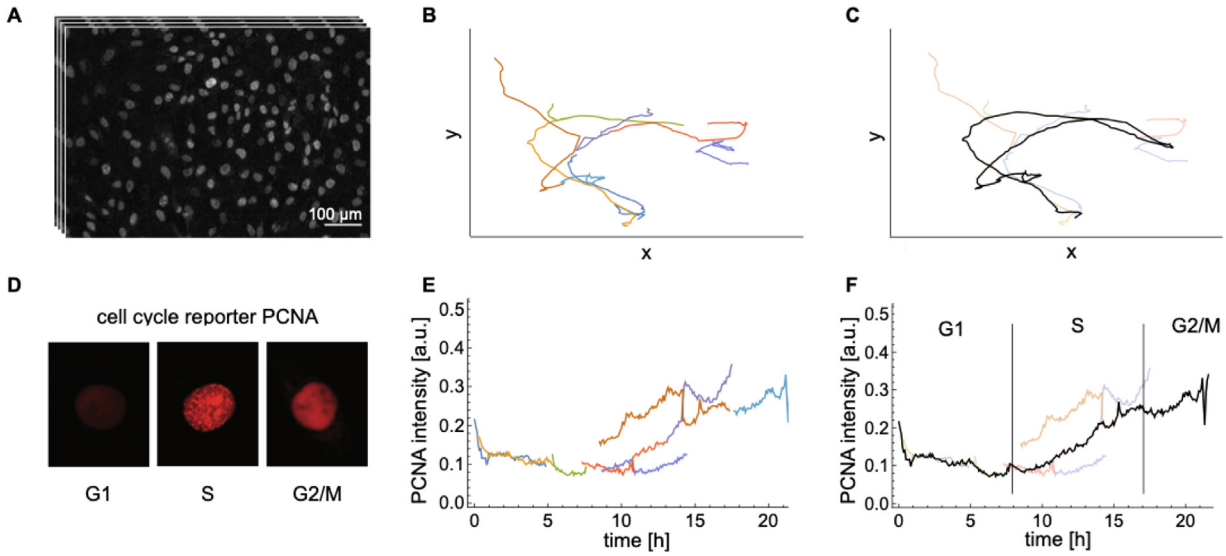

Fig. 3. (A) Example of time-lapse imaging data of retinal pigmented epithelial (RPE) cells using fluorescently tagged PCNA [32]. (B) The Fiji plugin IJTracker provides automatically generated short cell trajectories, denoted as AutoTracklets. (C) Using TraCurate, the AutoTracklets (coloured lines) can be efficiently combined and corrected to obtain accurate trajectories of RPE cells over complete cell cycles from one cell division to the next (black line). (D) PCNA was also used as an all-in-one marker to classify all cell cycle stages based on signal intensity and intracellular distribution (depicted are snapshot images in higher resolution). (E) The read-out of PCNA fluorescence intensity time-courses shown for AutoTracklets and (F) after correction with TraCurate. The single-cell time-courses over a complete cell cycle can then be used to determine cell cycle phases.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by funding of the Excellence Initiative by the German Federal and State Governments (Institutional Strategy, measure "support the best") to IR and the German Federal Ministry of Research and Education, Grant number 031A315 "MessAge" to IG. Open Access Funding by the Publication Fund of the TU Dresden, Germany.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.softx.2021.100656.

References

[1] Wallingford JB. The 200-year effort to see the embryo. Science 2019;365:758–9.
[2] Schroeder T. Imaging stem-cell-driven regeneration in mammals. Nature 2008;453:345–51.
[3] Skylaki S, Hilsenbeck O, Schroeder T. Challenges in long-term imaging and quantification of single-cell dynamics. Nat Biotechnol 2016;34:1137–44.
[4] McDole K, Guignard L, Amat F, Berger A, Malandain G, Royer LA, et al. In toto imaging and reconstruction of post-implantation mouse development at the single-cell level. Cell 2018. http://dx.doi.org/10.1016/j.cell.2018.09.031.
[5] Ulman V, Maška M, Magnusson KEG, Ronneberger O, Haubold C, Harder N, et al. An objective comparison of cell-tracking algorithms. Nature Methods 2017;14:1141–52.
[6] Maška M, Ulman V, Svoboda D, Matula P, Matula P, Ederra C, et al. A benchmark for comparison of cell tracking algorithms. Bioinformatics 2014;30:1609–17.
[7] Carpenter AE, Jones TR, Lamprecht MR, Clarke C, Kang IH, Friman O, et al. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. Genome Biol 2006;7:R100.
[8] Eliceiri KW, Berthold MR, Goldberg IG, Ibáñez L, Manjunath BS, Martone ME, et al. Biological imaging software tools. Nature Methods 2012;9:697–710.
[9] Tinevez J-Y, Perry N, Schindelin J, Hoopes GM, Reynolds GD, Laplantine E, et al. TrackMate: An open and extensible platform for single-particle tracking. Methods 2016. http://dx.doi.org/10.1016/j.ymeth.2016.09.016.
[10] Piccinini F, Kiss A, Horvath P. CellTracker (not only) for dummies. Bioinformatics 2016;32:955–7.
[11] Hilsenbeck O, Schwarzfischer M, Loeffler D, Dimopoulos S, Hastreiter S, Marr C, et al. fastER: a user-friendly tool for ultrafast and robust cell segmentation in large-scale microscopy. Bioinformatics 2017. http://dx.doi.org/10.1093/bioinformatics/btx107.
[12] Haubold C, Schiegg M, Kreshuk A, Berg S, Koethe U, Hamprecht FA. Segmenting and tracking multiple dividing targets using ilastik. In: De Vos WH, Munck S, Timmermans J-P, editors. Focus on bio-image informatics. Springer International Publishing; 2016, p. 199–229.
[13] McQuin C, Goodman A, Chernyshev V, Kamentsky L, Cimini BA, Karhohs KW, et al. CellProfiler 3.0: Next-generation image processing for biology. PLoS Biol 2018;16:e2005970.
[14] Amat F, Lemon W, Mossing DP, McDole K, Wan Y, Branson K, et al. Fast accurate reconstruction of cell lineages from large-scale fluorescence microscopy data. Nature Methods 2014;11:951–8.
[15] Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, et al. Fiji: an open-source platform for biological-image analysis. Nature Methods 2012;9:676–82.
[16] Schmidt U, Weigert M, Broaddus C, Myers G. Cell detection with star-convex polygons. 2018, ArXiv [CsCV].
[17] Fazeli E, Roy NH, Follain G, Laine RF, Chamier Lvon, Hänninen PE, et al. Automated cell tracking using stardist and trackMate. Cold Spring Harbor Laboratory; 2020, http://dx.doi.org/10.1101/2020.09.22.306233, 2020.09.22.306233.
[18] Hollandi R, Szkalisity A, Toth T, Tasnadi E, Molnar C, Mathe B, et al. nucleAIzer: A parameter-free deep learning framework for nucleus segmentation using image style transfer. Cell Syst 2020;10:453–8, e6.
[19] Lu AX, Zarin T, Hsu IS, Moses AM. YeastSpotter: accurate and parameter-free web segmentation for microscopy images of yeast cells. Bioinformatics 2019;35:4525–7.
[20] Rasse TM, Hollandi R, Horváth P. OpSeF IV: Open source python framework for segmentation of biomedical images. Cold Spring Harbor Laboratory; 2020, http://dx.doi.org/10.1101/2020.04.29.068023, 2020.04.29.068023.
[21] Berg S, Kutra D, Kroeger T, Straehle CN, Kausler BX, Haubold C, et al. ilastik: interactive machine learning for (bio)image analysis. Nature Methods 2019. http://dx.doi.org/10.1038/s41592-019-0582-9.
[22] Moen E, Borba E, Miller G, Schwartz M, Bannon D, Koe N, et al. Accurate cell tracking and lineage construction in live-cell imaging experiments with deep learning. 2019, 803205. http://dx.doi.org/10.1101/803205, BioRxiv.
[23] Moen E, Bannon D, Kudo T, Graf W, Covert M, Van Valen D. Deep learning for cellular image analysis. Nature Methods 2019. http://dx.doi.org/10.1038/s41592-019-0403-1.
[24] Bezanson J, Edelman A, Karpinski S, Shah V. Julia: A fresh approach to numerical computing. SIAM Rev 2017;59:65–98.

[25] Gonzalez-Beltran AN, Masuzzo P, Ampe C, Bakker G-J, Besson S, Eibl RH, et al. Community standards for open cell migration data. 2019, 803064. http://dx.doi.org/10.1101/803064, BioRxiv.
[26] Scherf N, Kunze M, Thierbach K, Zerjatke T, Burek P, Herre H, et al. Assisting the machine paradigms for human-machine interaction in single cell tracking. In: Bildverarbeitung für die medizin 2013. Springer Berlin Heidelberg; 2013, p. 116–21.
[27] Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, et al. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet 2000;25:25–9.
[28] Sluka JP, Shirinifard A, Swat M, Cosmanescu A, Heil RW, Glazier JA. The cell behavior ontology: describing the intrinsic biological behaviors of real and model cells seen as active agents. Bioinformatics 2014;30:2367–74.
[29] Diehl AD, Meehan TF, Bradford YM, Brush MH, Dahdul WM, Dougall DS, et al. The Cell Ontology 2016: enhanced content, modularization, and ontology interoperability. J Biomed Semant 2016;7:44.
[30] Burek P, Scherf N, Herre H. Ontology patterns for the representation of quality changes of cells in time. J Biomed Semant 2019;10:16.
[31] Burek P, Scherf N, Herre H. On the ontological foundations of cellular development. 2020, http://dx.doi.org/10.1101/2020.05.30.124875, BioRxiv.
[32] Zerjatke T, Gak IA, Kirova D, Fuhrmann M, Daniel K, Gonciarz M, et al. Quantitative cell cycle analysis based on an endogenous all-in-one reporter for cell tracking and classification. Cell Rep 2017;19:1953–66.
[33] Scherf N, Franke K, Glauche I, Kurth I, Bornhäuser M, Werner C, et al. On the symmetry of siblings: automated single-cell tracking to quantify the behavior of hematopoietic stem cells in a biomimetic setup. Exp Hematol 2012;40:119–30, e9.
[34] Schmid B, Shah G, Scherf N, Weber M, Thierbach K, Campos CP, et al. High-speed panoramic light-sheet microscopy reveals global endodermal cell dynamics. Nature Commun 2013;4:2207.
[35] Bach E, Zerjatke T, Herklotz M, Scherf N, Niederwieser D, Roeder I, et al. Elucidating functional heterogeneity in hematopoietic progenitor cells: a combined experimental and modeling approach. Exp Hematol 2014;42:826–37, e1-17.
[36] Rapoport DH, Becker T, Mamlouk AMadany, Schicktanz S, Kruse C. A novel validation algorithm allows for automated cell tracking and the extraction of biologically meaningful parameters. PLoS One 2011;6:e27315.