

## MICROSCOPY

## Improved tracking via cell region fingerprints

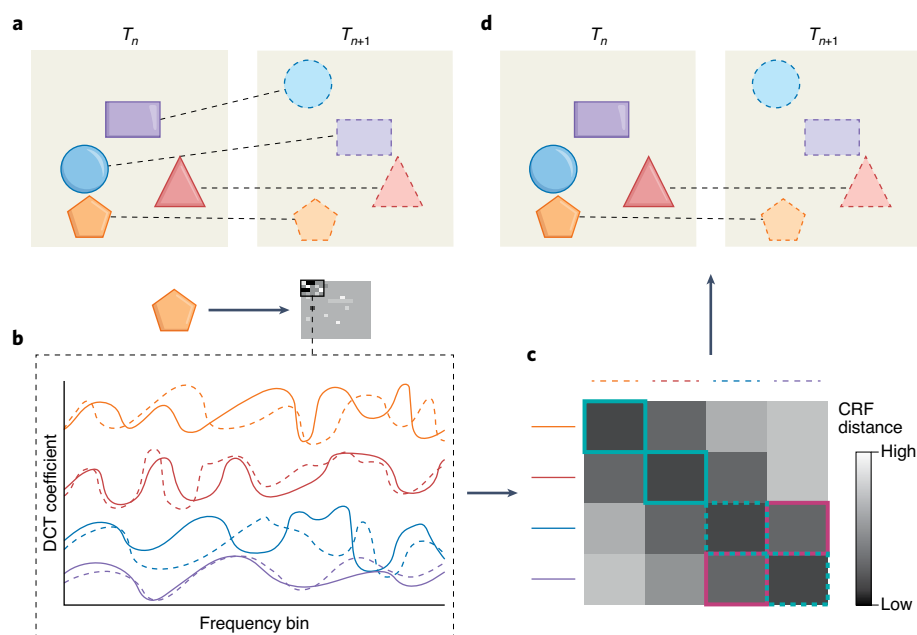
TracX shows that fast proofreading of track assignments leads to improved cell tracking performance.

Beth A. Cimini

Object tracking is one of the most notoriously difficult problems in microscopy image analysis. Tracking in bioimages typically involves a large number of nearly identical objects that are tightly clustered and may merge or divide many times. Since nearly all tracking methods rely on first detecting or segmenting objects, the quality of object identification needs to be extremely high; also, many tracking algorithms require iterative tuning of many parameters to perform well. It is therefore unsurprising that bioimage analysis tool users consider tracking one of the least well solved problems in bioimage analysis<sup>1</sup>. Manual curation of ground truth for tracking experiments is slow and painful; while a few excellent benchmarks exist<sup>2,3</sup> and are useful for testing algorithm performance in general, users who want to know how well a given method and parameters work on their own particular data often have little choice other than to create this ground truth or, more commonly, assess a few sets by eye and hope that they are representative.

In this issue, Cuny et al.<sup>4</sup> release the open-source software TracX in an attempt to solve two of the major problems in this space: the need for a tool that does not require hours of iterative parameter tuning to perform well and the need to quantitatively assess tracking performance without hand-creating ground truth for one's own dataset. TracX users must first create segmentations in either the authors' CellX<sup>5</sup> tool or any tool that can output a binary mask or a label matrix. When raw images and corresponding segmentations are loaded into the TracX graphical user interface, the user need only provide the list of frames to analyze and a single initial tunable parameter: whether the objects to be tracked are asymmetrically dividing yeast or bacteria, symmetrically dividing yeast or bacteria, or mammalian cells.

Tracking in TracX begins as in many other tools, via creation of proposed tracks based on minimization of a cost function balancing object displacement, size deviation, rotation, and number of frames skipped (if any); from here,



**Fig. 1 | Cell tracking with TracX.** **a**, For each object at each time point, a tentative track assignment is made based on a cost function taking into account cell displacement, cell size deviation, cell rotation and frame skipping. **b**, Each object's image is run through a discrete cosine transform (DCT); the lowest frequencies are taken as a cell region fingerprint (CRF). **c**, The frequency-space distance between all objects at  $T_n$  and  $T_{n+1}$  is computed. Solid lines indicate the assignment based on **a**, dashed lines the best possible match. **d**, If a high enough fraction of the other  $T_{n+1}$  objects have smaller distances to the  $T_n$  object than the  $T_{n+1}$  object it was initially matched to, the latter assignment is deleted, allowing remapping at a later stage.

TracX adds a trick from the world of data compression. After cropping the input image to a loose bounding box around each object, it scales it, applies a discrete cosine transform (DCT)<sup>6</sup> to the cropped images and subsamples only the lowest frequencies (which carry the structural information but not fine details) to create what the authors call the 'cell region fingerprint' (CRF). The difference in CRF between objects (the 'CRF distance') can be measured for each object in time point  $n$  ( $T_n$ ) to some or all objects in the next time point ( $T_{n+1}$ ) (Fig. 1a,b). By comparing the CRF distances, one can assess for each  $T_n$  object whether the  $T_{n+1}$  object it was initially matched to has the highest image structural similarity (and therefore the lowest CRF distance)

or not; for each  $T_n$  object, if more than a specified fraction of the nearby  $T_{n+1}$  objects have a lower CRF distance than its initial assignment, that assignment is thrown out and the objects go through future rounds of cost-function-based tracking (Fig. 1c,d).

This proofreading step massively improves performance: while without proofreading TracX's tracking performance was 30–60% across four varied image sets, requiring proposed tracks to be in the best 20% of CRF distances for acceptance produces nearly perfect performance on these sets, compensating even when up to half of the initial assignments have been intentionally shuffled. The authors show that the four proofreading parameters (bounding box size, size the bounding box is

resized to, number of DCT values included in the CRF, and fraction of neighbors with lower CRF distances needed to remove a track) either minimally affect performance or have default value ranges that work for a wide range of image types. In addition to the performance boost from proofreading, the fraction of initial assignments removed serves as an ideal measurement of initial tracking quality, giving users a quantitative way to optimize the tuning of the initial cost function parameters without handmade ground truth. The rejected assignments can also be directly examined to discover tracking failure modes or even segmentation errors, which the authors found proofreading identified with high accuracy.

While this exciting advance could in theory be applied in any tool that performs tracking, the TracX program itself has a number of other nice features to recommend it: fast performance; the ability to use your favorite segmentation algorithm; the ability to filter objects by size, position or additional fluorescence channels; support for multiple division types; and even lineage tracing. This is not to say that TracX is perfect; the choice

of MatLab decreases direct interoperability with popular Python-based segmentation tools like Cellpose<sup>7</sup> and CellProfiler<sup>8</sup> and may make transition to a non-graphical user interface harder for users used to those tools or the Java-based TrackMate<sup>9</sup> for tracking. Most of the reported benchmarking is on yeast data taken at short time intervals, making it unclear how TracX will perform on most non-yeast data or less densely sampled time-lapse movies. Benchmarks on and cost function parameters for more or all of the Cell Tracking Challenge<sup>2</sup> datasets would be a valuable community contribution. Finally, it is not clear how separable segmentation and tracking will be or should be in the future, as new deep learning approaches may combine both segmentation and tracking into single networks<sup>10</sup>. Whatever the future holds, TracX performs very well with approaches available right now and is undoubtedly a step forward in making high-quality tracking easy for novice bioimage analysts. □

Beth A. Cimini    
Imaging Platform, Broad Institute of MIT and

Harvard, Cambridge, MA, USA.

✉e-mail: [bcimini@broadinstitute.org](mailto:bcimini@broadinstitute.org)

Published online: 22 September 2022

<https://doi.org/10.1038/s41592-022-01525-z>

#### References

1. Jamali, N., Dobson, E. T. A., Eliceiri, K. W., Carpenter, A. E. & Cimini, B. A. *Biol. Imaging* **1**, e4 (2022).
2. Ulman, V. et al. *Nat. Methods* **14**, 1141–1152 (2017).
3. Versari, C. et al. *J. R. Soc. Interface* **14**, 20160705 (2017).
4. Cuny, A. P., Ponti, A., Kündig, T., Rudolf, F. & Stelling, J. *Nat. Methods* <https://doi.org/10.1038/s41592-022-01603-2> (2022).
5. Mayer, C., Dimopoulos, S., Rudolf, F. & Stelling, J. *Curr. Protoc. Mol. Biol.* **14**, Unit 14.22 (2013).
6. Ahmed, N. *Digit. Signal Process.* **1**, 4–5 (1991).
7. Stringer, C., Wang, T., Michaelos, M. & Pachitariu, M. *Nat. Methods* **18**, 100–106 (2021).
8. Stirling, D. R. et al. *BMC Bioinformatics* **22**, 433 (2021).
9. Ershov, D. et al. Preprint at *bioRxiv* <https://doi.org/10.1101/2021.09.03.458852> (2021).
10. Löffler, K. & Mikut, R. Preprint at *arXiv* <https://doi.org/10.48550/arXiv.2204.10713> (2022).

#### Acknowledgements

This work was supported by the National Institutes of Health grant P41 GM135019 as well as grant number 2020-225720 from the Chan Zuckerberg Initiative DAF, an advised fund of the Silicon Valley Community Foundation.

#### Competing interests

The author declares no competing interests.