Ctc722 (SIAT-CN) Cell Tracking Method

Authors: Zhike Zi
Contact: zk.zi@siat.ac.cn
Platform: Compatible with Mac/Linux/Windows
Prerequisites: The current version is MATLAB-based, requiring MATLAB and the following toolboxes:

- MATLAB Image Processing Toolbox
- MATLAB Parallel Computing Toolbox
- matlabPyrTools (included)

Summary: Ctc722 (SIAT-CN) presents a cell tracking method utilizing an optimized score that combines Euclidean distance between cells and their neighborhood similarity. This parameter-free approach enables simultaneous tracking of multiple cells over time. While currently implemented in MATLAB, future plans include releasing the algorithm in Python and/or as an ImageJ plugin.

Preprocessing: We first estimate cell movement within the chosen frame using the classic Euclidean distance approach, comparing the centroid locations of cells over consecutive frame. This movement distance is defined as the Euclidean distance between the current frame's cell centroid and the nearest neighboring cell centroid from the previous frame. In addition, we compute diameter and circularity for all segmentation masks in the selected frames. To optimize the preprocessing time, we only select the middle and last frame for preprocessing analysis.

Circularity information aids in automatically distinguishing cell segmentations as either nucleus or nucleus-like objects. For nucleus or nucleus-like segmentations, we define the maximum tracking distance.

```
MAX_TRACK_DISTANCE = ceil(2*diameter_median + 2*cell_movement_CI99_UB);
```

Otherwise:

```
MAX_TRACK_DISTANCE = ceil(1.5*diameter_median + 3*cell_movement_CI99_UB);
```

Here, cell_movement_CI99_UB represents the upper 99% confidence bound value of cell movement distance, and diameter_median denotes the median diameter of segmentation masks.

Cell Tracking (Cell Linking Strategy): Our tracking algorithm introduces an integrated tracking index score (zTrack_index) for cell *i* in current frame t and cell *j* in previous frame t-1, comprised of two components:

1) Euclidean distance (d_j^i) between the centroid of the cell *i* in current frame t and the centroid of the cell *j* in previous frame t-1.

2) A similarity score (s_j^i) of between the cropped images centered with the centroid of the cell *i* in current frame t and the centroid of the cell *j* in previous frame t-1 (see the image below). The details about the similarity score will be provided in the future (manuscript in preparation).

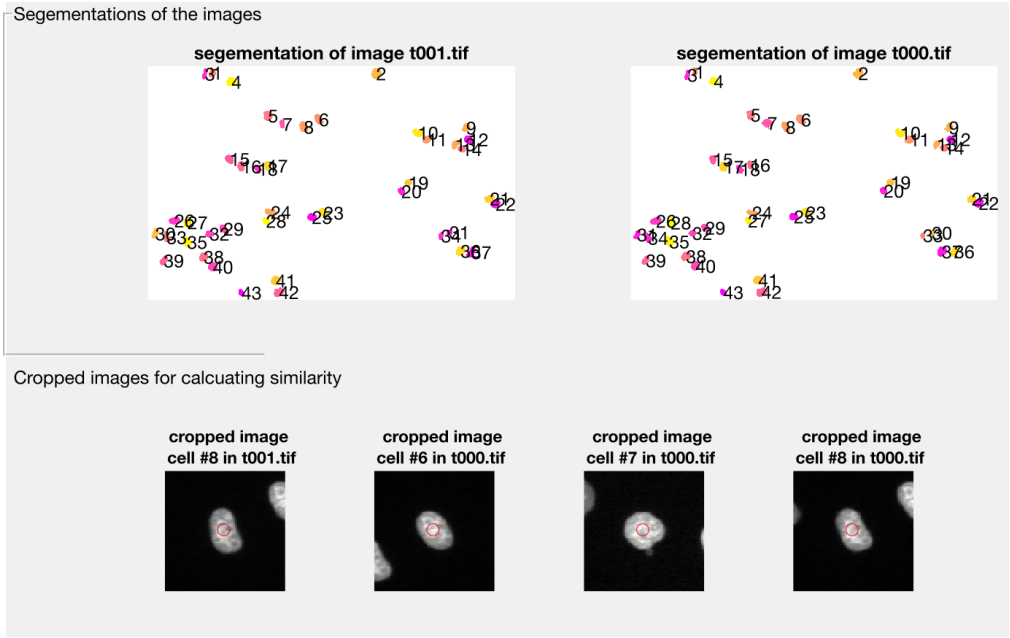

Illustration of cropped images for cell *i* in frame t and other cells *j* in frame t-1, the cropped image size is dynamically assigned based on the diameter size of cell *i* . The red circles indicate the centroid of cell segmentations in current frame and previous frame (dataset: Fluo-N2DL-HeLa/01)

For each cell pair (cell *i* in frame t and cell *j* in frame t-1), we calculate the zTrack_index score with the following equation only if $d_{i,j}$ <= MAX_TRACK_DISTANCE

$$zTrackIndex_j^i = s_j^i - w * d_j^i$$

Here, *w* is a penalty weight for Euclidean distance (d_j^i).

if $d_{i,j}$ > MAX_TRACK_DISTANCE, the corresponding tracking score is NaN.

We link current cell *i* in frame t to the cell *k* in frame t-1 if the corresponding $zTrackIndex_k^i$ is the maximum value among all $zTrackIndex_j^i$ values, and if it is larger than 0. If the maximum value of $zTrackIndex_j^i$ is less than 0, we consider cell *i* as a newly appeared cell without mother cell in frame t-1.

In essence, the zTrackIndex comprehensively evaluates each cell pair by incorporating both their Euclidean distance and the similarity between their cropped images, which mirrors our eye-tracking method, looking at neighboring environment surrounding the paired cells.