U-Track Documentation

Release 2.1.0

LCCB

August 23, 2013

CHAPTER

ONE

GENERAL INFORMATION

1.1 License

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see http://www.gnu.org/licenses/.

1.2 Requirements

The program has been tested using the following configuration(s):

Operating systems: Ubuntu 10.04 64-bit, Mac OS X 10.7.5 64-bit, Windows 7 64-bit.

Matlab version: 2011a and above

Matlab toolboxes: to run ALL components of the software, the following Matlab toolboxes MUST be installed:

– Curve Fitting Toolbox. – Image Processing Toolbox, – Statistics Toolbox, - Optimization Toolbox,

1.3 Installation

1. Download the latest version of the software from http://lccb.hms.harvard.edu/software.html.
2. Extract all the files from the zip file that was downloaded.
3. Start Matlab
4. Add the directory containing the code to the Matlab path

1.4 Changes

Modifications of U-Track since the publication of the paper (Jaqaman et al., Nat. Methods 2008):

2.1 (June 2013)

- Fix graphical interface bug in Matlab R2013a and above. Thanks to Riccardo Felletti and Liam Holt for reporting this issue.
- Allow the analysis of movies stored on an OMERO image database. Analysis results are uploaded onto the OMERO server as a file attachment.

2.0 (March 2013)

- Include two new tracking applications as part of the U-Track package: microtubule plus-end tracking (previously distributed as plusTipTracker) and nuclei tracking.
- Implement a third optional processing step to the analysis workflow: track analysis with two methods: motion analysis (see Jaqaman et al. Cell 2012) and microtubule plus-end classification (see Applegate et al. JSB 2011)
- Allow the preselection of the tracking application when running U-Track for the first time on a movie. Detection, tracking and track analysis default parameters/methods/cost matrices are initialized with respect to the chosen tracking application.

21 December 2011

- Tracking code modifications to improve speed: The tracking code has been internally modified to enhance its speed; it runs faster and no longer slows down almost exponentially with increase movie length. No effect on input or output.
- New cost functions costMatRandomDirectedSwitchingMotionLink and costMatRandomDirectedSwitchingMotionCloseGaps: These new cost functions are similar to their predecessors with one added motion model option, namely moving in a directed manner in a certain direction without the possibility of immediate direction reversal as was the case before. These cost functions have 3 options for motion propagation (instead of 2):
  - linearMotion = 0: one motion model, namely random (Brownian) motion.
  - linearMotion = 1: two motion models, namely random motion and movement with constant velocity.
  - linearMotion = 2: two motion models, namely random motion and movement along a straight line but with the possibility of immediate direction reversal.

  What was linearMotion = 1 in the previous cost functions corresponds to what is linearMotion = 2 in the new cost functions. Everything else is the same.

  Examples: Motor-driven movement is best tracked with linearMotion = 1. One-dimensional diffusion is best tracked with linearMotion = 2. Diffusive movement without any drift or directionality is best tracked with linearMotion = 0.

1 April 2011

- Detection – using absolute background information: The detection code can take additional input arguments that supply it with images of "absolute background" and an alpha-value to compare local maxima to this absolute background.

  "Absolute background" is usually a cropped subpart from the original images, where the cropped area lies outside of the cell. Since this area tends to be quite dimmer than inside the cell, one can use a stricter alpha-value, for example 0.001, to compare local maxima to this background area. The use of a stricter alpha-value for comparison with background outside of the cell and a less strict alpha-value for comparison with local

background inside the cell minimizes false positives outside of the cell AND false negative inside the cell, where the objects of interest are located.

If this option is used, there must be an absolute background image for each original image.

- costMatLinearMotionCloseGaps2 – explicit definition of power for scaling search radius with time:

  In the previous version of costMatLinearMotionCloseGaps2, the user only defined "timeReachConfB" and "timeReachConfL." Given these parameters, the code internally scaled the search radius with the square root of time before "timeReachConfB" and "timeReachConfL," and with time^0.01 after.

  In the new version, the user defines these powers explicitly, to give more flexibility. Setting the new parameters brownScaling and linScaling to [0.5 0.01] is equivalent to the old settings.

25 April 2010

- Detection code bug fix: The previous version of the detection code was overestimating the position uncertainties by a factor of (PSF sigma)^2. u-track modifications since the publication of Jaqaman et al. Nat. Methods 2008
- Code modifications to handle larger datasets: Both the detection and tracking codes have been modified to better handle larger datasets. This has no effect on their input or output though.
- New cost functions costMatLinearMotionLink2 and costMatLinearMotionCloseGaps2:

  These new cost functions are similar to their predecessors, with a few "behind-the-scenes" changes.
- Distance cost scaling in costMatLinearMotionCloseGaps2: For gap closing, merging and splitting, the part of the cost based on distance is now scaled by the average frame-to-frame displacement of the track segments involved. This avoids punishing particles that are more mobile relative to those that are less mobile.
- Alternative costs in costMatLinearMotionCloseGaps2: Previously, the gap closing alternative cost was taken as the 90th percentile of the costs of all potential assignments. Furthermore, the merging and splitting alternative cost was determined on a case-by-case basis. In the new cost function, all alternative costs are assigned the same value, taken as the X percentile of the distribution of potential assignment costs, where X is calculated from the structure of the matrix of potential assignments (in particular, it takes into account the number of potential assignments each track segment has).
- Auxiliary (lower right) block costs in costMatLinearMotionLink2 and costMatLinearMotionCloseGaps2: Those were previously assigned to be the smallest costs in the cost matrix, the goal being that they should not influence the LAP outcome. But, in retrospect, having the lowest costs might favor them, thus influencing the LAP outcome. Thus, they have been modified to be equal to the alternative costs, so that they are truly neutral and do not influence the LAP outcome.

26 June 2009

- Diagnostics:
  1. Gap closing time window: If the additional field "gapCloseParam.diagnostics" is set to 1, the software will plot in the end of tracking a histogram of the closed gap lengths. This will help with assessing the quality of gap closing. Generally speaking, longer gaps should be less frequent than shorter gaps; thus, a gap length histogram with a plateau might be indicative of a too large gap closing time window.
  2. Maximum search radius: The frame-to-frame linking cost function has the additional input "parameters.diagnostics", through which the code will output histograms of frame-to-frame linking distances at the specified frames. For example, if parameters.diagnostics = [2 35], then the histogram of linking distances between frames 1 and 2 will be plotted, as well as the overall histogram of linking distances for frames 1->2, 2->3, ..., 34->35. The histograms can be plotted at any frame except for the first and last frame of a movie. To not plot, enter [ ].

- Gap length penalty: The gap closing cost function has the additional input "parameters.gapPenalty" to penalize longer gaps. If parameters.gapPenalty has the value p, then the penalty for a gap of length p will be p^n. Note that for p > 1 longer gaps are penalized, for p = 1 there is no gap length penalty, while for p < 1 longer gaps are favored.
- Resolution limit: The gap closing cost function has the additional input "parameters.resLimit", representing the resolution limit in pixels. The resolution limit is generally the Airy disk radius, but it could be smaller when iterative Gaussian mixture-model fitting is used for detection. The resolution limit is used to expand the merge/split search radius if it is found to be smaller than the resolution limit.

12 December 2008

- An additional Kalman filter function (default: kalmanReverseLinearMotion) to time-reverse Kalman filter information for the second and third rounds of frame-to-frame linking.

CHAPTER

TWO

MOVIE SETUP

2.1 Movie selection

1. From the Matlab command prompt, launch the movie selector interface by typing:

   ```
   >> movieSelectorGUI
   ```

   This command will bring up the movie selection panel (Fig. *Movie selection panel*). The left panel displays all of the movies to be processed. The buttons next to the list allow the user to modify the list by creating, opening, and removing movies. The right panel displays the available software packages that can be used to process the movies.

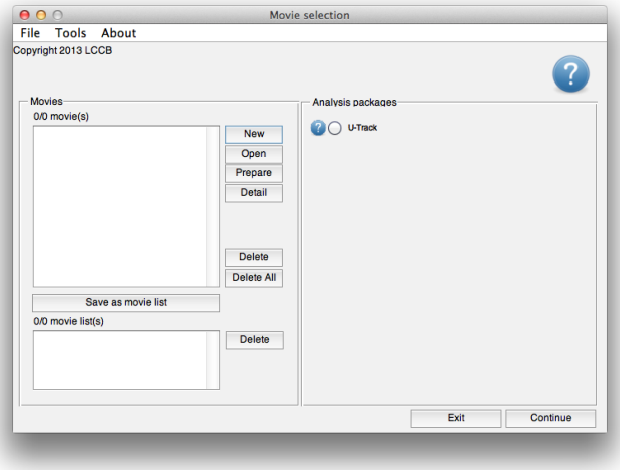

Figure 2.1: Movie selection panel

2. Add a movie to the processing list by either creating new a new movie database (see *Movie creation* section below) or loading an existing movie saved on disk (see *Movie loading* section below). Repeat until all of the movies to be processed are listed in the left panel.

   A list of movie databases can be saved as a list using the Save as Movie List button. The resulting movie list is also saved as a MAT file on the disk. To load all of the movies in a list, repeat step 14 and select the MAT file containing the movie list.

3. Select the package on the right panel and click *Continue* at the bottom right of the movie selection window.

   This will bring up the analysis panel.

2.2 Movie creation

There are two ways to create movie databases: from series of TIFF files per channel (see section *Movie creation using series of images files* below) or directly from proprietary files using Bio-Formats (see section *Movie creation using Bio-Formats* below).

2.2.1 Movie creation using series of images files

1. Prepare the image files for the database by storing each channel (wavelength) of each time-series in a separate directory (folder), with one file per frame (time point) of the movie. To indicate the time point for the software package, use the following filename convention: MyMovieXXX.tif.

   `MyMovie` is a placeholder for any string, including underscores and dashes that specifies the generic name of the movie. XXX is a placeholder for a numeric value identifying the time point of that particular frame, e.g., 007, 008, 009, etc.

2. In addition to organizing the raw image sequences, gather information on the camera (pixel size, acquisition time, bit depth), the objective lens (numerical aperture), and the fluorescent channels (emission wavelength) before launching the software.

3. Create a new movie database by clicking on *New* from the *Movie selection panel*.

   This will bring up the movie edition interface (Fig. *Movie edition interface*).

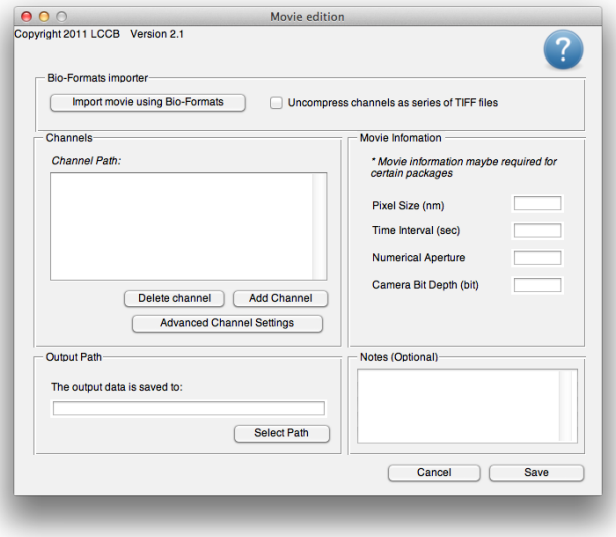

Figure 2.2: Movie edition interface

4. Click on *Add Channel*. Select the folder containing the images to be analyzed. Repeat the *Add Channel* operation for all of the channels of the movie that is to be analyzed.

5. Click on *Advanced Channel Settings*. In the new window (figure *Channel edition interface*), fill out the emission wavelength (in nm) for each channel.

6. In the movie edition interface, fill in the pixel size (in nm), time interval (in sec), numerical aperture, and camera bit depth of the movie.

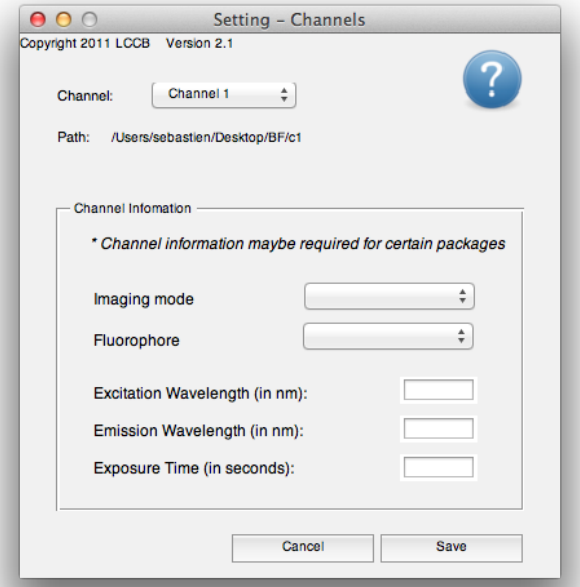

Figure 2.3: Channel edition interface

Once saved, these fields cannot be further modified. In case of an erroneous input, a new movie database must be generated by repeating all the steps above.

7. Optionally, enter additional notes specific to the movie.
8. In the output directory panel, click Select Path and choose a location on the disk where all results of the analysis should be saved.
9. Click on *Save*.

   This will open a pop-up window asking where to save the movie database. The operation will save a MAT file (Matlab format) containing all the movie information including all results from the processing. This MAT file can be later reused for loading the movie database (see next section).

2.2.2 Movie creation using Bio-Formats

Alternatively, movie databases can be directly created from image files using Bio-Formats.

1. Click on *Import Movie using Bio-Formats* and select the file containing the movie. This will automatically read the metadata if present. It also allows the reading of image sequences from image stacks and proprietary image formats, avoiding preparatory steps above.

   For further information, click on the Help button in the upper right corner of the window, which will launch the online documentation for the software.
2. Optionally, enter additional notes specific to the movie and missing metadata.
3. Click on *Save*.

2.3 Movie loading

From the movie selection panel (Fig. *Movie selection panel*), load the new movie by clicking *Open*. Select the MAT file saved when creating the movie database.

If the movie database file has been relocated on the disk, the software will ask for relocation of all its components by comparing the new path of the movie database file to the old path.

CHAPTER

THREE

ANALYSING OMERO MOVIES

Since version 2.1, U-Track is able to analyze movies stored on an OMERO image database.

Warning: Setting up OMERO movies for analysis using U-Track is only achievable via the command-line currently but some graphical interface improvements will be added to the software to simplify this step.

Below is a list of steps to load and analyze OMERO movies using U-Track:

1. Download and install the OMERO.matlab toolbox corresponding to your OMERO server as described on the OMERO.matlab documentation page.
2. Create an OMERO session using the server name, user name and password,

   e.g.:

   ```
   client = loadOmero('my-server');
   omeroKeepAlive(client);
   session = client.createSession('my-user', 'my-password');
   ```

   ```
   For more information, refer to the 'OMERO.matlab documentation page'_.
   ```

1. From any OMERO client (OMERO.insight or OMERO.web), make a list of the identifiers of the movies you want to analyze.
2. Load the movies to analyze using:

   ```
   MD = MovieData.load(session, ids)
   ```

   For each movie, this function will first check if U-Track has been run and the analysis attached to the movie onto the OMERO.server. If that is the case, the previous analysis will be reloaded and reused else a new movie analysis database will be created.

3. Load the U-Track interface:

   ```
   uTrackPackageGUI(MD);
   ```

CHAPTER

FOUR

U-TRACK

4.1 Overview

The tracking software can be launched by selecting the *Tracking* in the movie selection interface. If the tracking software has already been run on *all* the movies, the main analysis panel will show up. Else a dialog box will open (Figure *Tracking initialization dialog box*) asking the type of objects to be tracking.

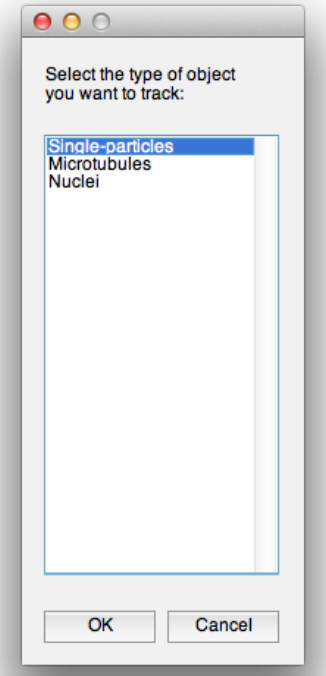

Figure 4.1: Tracking initialization dialog box

Tracking is currently implemented for 3 kinds of applications: single-particles (2D), microtubules (2D) and nuclei (2D). Detection, tracking and track analysis default parameters will be determined based on the application choice.

4.2 Workflow

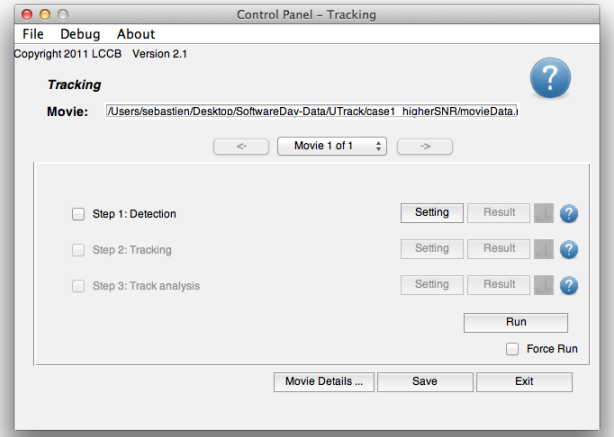

Figure 4.2: Main panel of the U-Track analysis software

The U-Track workflow currently consists of three consecutive analysis steps: detection, tracking and analysis. In accordance with this analysis work flow, check the components of following main panel of the U-Track software (Fig. *Main panel of the U-Track analysis software*):

1. A top panel that displays the current processed movie. If processing multiple movies, use the drop-down menu/arrows to switch between movies.
2. A sequence of processes that can be run either individually or in a complete flow.

   Each process needs to be set up before it can be run. To set up a process, click on its associated *Setting* button. Once set up, processes appear in bold characters.
3. A checkbox next to each process marks it as scheduled for processing.

   To run a process, check the respective box and click *Run*. Once successfully run, the output of the process can be visualized by clicking the *Result* button.
4. Icons next to each process indicate the current processing status. Only processes that have been run at least once successfully have an associated icon. To obtain more information, click on an icon.
5. A *Run* button. Only checked processes with non-green icons will be run when hitting *Run*. If a process that has been run before needs to be re-ran, check the *Force Run* box.

In the case of processing multiple movies through multiple steps, the processes that are to be run need to be individually set up and checked.

In each setting interface, check the *Apply to All Movies box* to set up processes in batch. Check *Apply Check/Uncheck to All Movies* to schedule processes to run for all movies. Finally, click *Run All Movies* to run scheduled processes of all movies in the list.

All interfaces of the U-Track have a Help icon located in the top right corner of the window. Click on the icon to open the help document as a PDF file.

Each analysis step will be prepopulated with default values depending on the kind of tracking application the package was initialized with.

4.3 Single-particles

Default choices are to use:

- the Gaussian Mixture-model fitting as the detection method,
- the Brownian and directed motion linking cost matrix
- the Brownian and directed motion gap closing cost matrix
- the Motion analysis post-tracking step.

4.4 Microtubules

Default choices are to use:

- the Comet detection as the detection method,
- the Microtubule plus end linking cost matrix,
- the Microtubule plus end gap closing cost matrix,
- the Microtubule dynamics classification post-tracking step.

4.5 Nuclei

Default choices are to use:

- the Nuclei detection as the detection method,
- the Brownian and directed motion linking cost matrix
- the Brownian and directed motion gap closing cost matrix
- the Motion analysis post-tracking step.