

```

% AUTHOR: Mustafa Elmas
%
% DATE: 08/01/2017
%
% AFFILIATION: University of Tennessee - Knoxville
%
%
% PURPOSE:
%
% To analyze bacterial motility videos and find
% I) Cell speed (micron/s)
% II) Cell reversal frequency (/s)
% III) Mean square displacement (MSD)
%
% 1. Split a recorded video into certain number of frames
%
% 2. Convert the frames into binary frames
%
% 3. Calculate Centroid, Major and Minor axis length and angle
% by MATLAB built-in function, regiongroup.
%
% 4. Constructs n-dimensional trajectories from a scrambled
% list of particle coordinates determined at discrete times
% in consecutive video frames by MATLAB version of cell-tracking
% algorithm by Crocker and Grier.
%
% 5. Trajectories slower than 10 microns/second and shorter than 1 s were
% excluded from the analysis. This ensures that we restrict our
% analysis mostly to trajectories that lie in a narrow zone around
% the focal plane.
%
% 6. Calculate velocity, reversal frequency, acceleration, angular
% acceleration, velocity autocorrelation, Mean Square Displacement

```

Prerequisite for the best code to work:

1) Matlab version R2019a – The version of MATLAB is important since the syntax of MATLAB may be updated due to version upgrade. Using the code in another version of MATLAB might result in unmatched syntax that will lead to errors.

2) MATLAB has to have the **image processing toolbox, statistic toolbox, and mapping toolbox**.

For instruction on adding toolbox,

https://www.mathworks.com/help/matlab/matlab_external/support-package-installation.html#:~:text=You%20install%20support%20packages%20using,Ons%20%3E%20Get%20Hardware%20Support%20Packages.

3) A **30 seconds (any more duration than this will result in disordered combinatorics in the tracking process that will result in shorter tracks)** video for analysis has to be in good quality for quality analysis (meaning bad video = bad data). The heuristics of a good video are clear visuals of moving bacteria, the population of cell is not too dense (20-30 cells per video should be the number of cells that one shoots for), interrupted frames, and no environmental

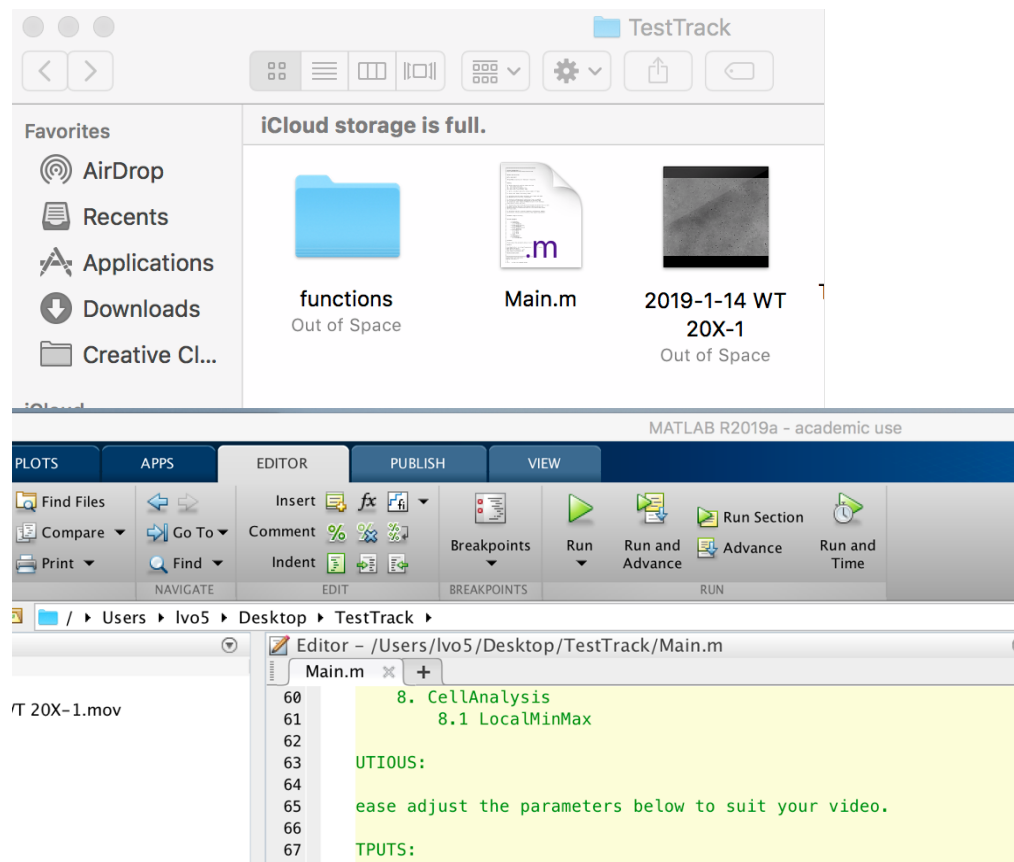
movements (no shaking or mass movement from suspensions). A good video is provided as an example (**TM000437.MP4**).

For any questions about the codes, please contact Mustafa Elmas (melmas@vols.utk.edu),
Lam Vo (lvo5@vols.utk.edu)

For any questions about the science, please contact Gladys Alexandre (galexan2@utk.edu),
Tanmoy Mukherjee (tmukherj@vols.utk.edu)

How to use tracking code:

- 1) Download the code zip files and extract into a specific folder
- 2) In the folder you will see: You need to add the video you want to analyze in that folder – the code only run one video at a time
- 3) Open the Main.m matlab code. For these next steps, I suggest running each section one by one so it's easier to keep track of what is going on. You can do that by pressing run and advance. At the Initialize variable section, set the pixel/micron ratios and change in time based on FPS.



- 4) Click run and advance until you yellow highlights this box. You need to change the VideoName variable to the name of the video you want to analyze. The Vtype is the type of video (avi, mp4,...)- **I have the best results using raw MP4 files from recording. I have bad experiences with using .avi files because it creates repeated frames, which is detriment the analysis.**

```
112 SmoothMETHOD = InitValues();
113 %% We initialize the variable manually here!
114 - VideoName = '2019-1-14 WT 20X-1'; %Put the video name here
115 - Vtype = 'mov'; %type of video
116 - windowWidth = 10; %change the window of track separation to have
117 %% Add track Videos folder
```

- 5) Run again until you reach. Make sure you change the OriginalSubfolder to the framepath that has the video. For the Video2Frames function, you need to put in the name of the video that you want to analyze as the first argument.
- 6) Run the 'Extract frame' section. I will ask you if you want to save your frames to another out of disk folder. I put yes.
- 7) After the code done extracting the frames, the code will put a new folder into your current one. **You need to go to that Users folder and cut the frames and move it to the folder with the videos name. Then, you need to change the name of the folder, the one that you just move the frame into, to the name of the video without the extension.**
- 8) Run the 'Detect cells' section. This will take a lot of time (15-20 min) and this process will take a lot of your computer's processing power. It is suggested that you try not to do anything else on your computer while running this section. It will be done when the variable DataPosition is in the workspace.
- 9) Run the 'Tracking cells' (also takes a while) and then the 'Smooth tracks' sections.
- 10) Run the "CELL ANALYSIS BEFORE DELETING BAD TRACKS" section.
- 11) Run "PLOT TRACKS BEFORE DELETION". This will result in a series of pop-up MATLAB windows that shows the tracks of the cells. This is incredibly helpful for spotting tracks that are not good for further analysis (for example, if you are analyzing a chemotactic mutant, you would want to distinguished between tracked cells that are moving and cells that are not moving (and is subjected to Brownian motions and fluid movement – usually erratic, small length movement might indicate that cell is not actually motile or moving). **Write down the track ID numbers that is bad and track ID numbers that are good examples. After you are, done type "close all" at the command window to close all the figure.**
- 12) Run "DELETE UNWANTED TRACKS" section. On the command window, it will ask you where you want to delete any tracks. Type Y and enter for yes. And Type N and enter for no. If you put yes, it will ask you what track ID you want to delete. You will have to type in the track ID of bad tracks in form of [1,2,3, ...] and enter.
- 13) Run "CELL ANALYSIS AFTER DELETING BAD TRACKS" section. Optionally, you can then run "PLOT TRACKS AFTER DELETION" to see any leftover bad tracks you want to delete. You would have to go back to step 12 to delete any more tracks.
- 14) Continue to run the "save dataCLEAN" , "Load the data", and "Write in excel" sections. This will create an excel file in your main folder called dataCLEAN, which consists of a raw datasets of tracks. X-coordinate, Y-coordinates, Velocity, Time, and Cell ID will be given.

15) To produce video with super-imposed tracked for publication purposes, run the “Produce Individual tracks”, “IMPOSED TRACK ON FRAMES”, “VIDEO OF INDIVIDUAL TRACK” sections. For the “Produce Individual tracks”, be sure to change the TvideosPath argument to the file path of your video that you are analyzing (if in the same folder, just replace the video file name in this argument). For IMPOSED TRACK ON FRAMES section, change the TrackNum argument into the number of track ID you want a video of (you should have this at step 11).

16) To calculate cell track statistic, run the “Cell Track Statistic section”.

17) Now to export the statistics of the tracks (specifically, reversal angle, speed to reversal ratio, run time before reversal, angular velocity, average velocity, reversal frequency, and tumble duration, and mean square displacement), run the “EXPORTING STATISTICS” section. Note that you have to change the directory on the writetable argument that has the new excel file name (that you pick) for the section to work.

Notes:

1) Your computer does not have enough space to store every file that is provided by the code. It would be wise to save the necessary videos and picture from tracking. To back-up your statistical and tracking analysis, it would be best to save the workspace from MATLAB after you are done with the analysis. You can load the workspace to redo any statistical analysis if you have to.

2) The frames stored in the main files take a lot of computer space. It would be best to remove all the videos frames after analysis to start new one.

3) It is suggested that you rename the excels and video files after you are done with the analysis and move it to another folder. This will ensure that the code will not mistake any previous analysis as variables for future analysis.