TIAM: the Tool for Integrative Analysis of Motility

Getting started

To run TIAM, the user needs to hold MATLAB license with image processing signal processing tool boxes. Java Run time Environment must also be installed. Both MATLAB and Java are platform independent.

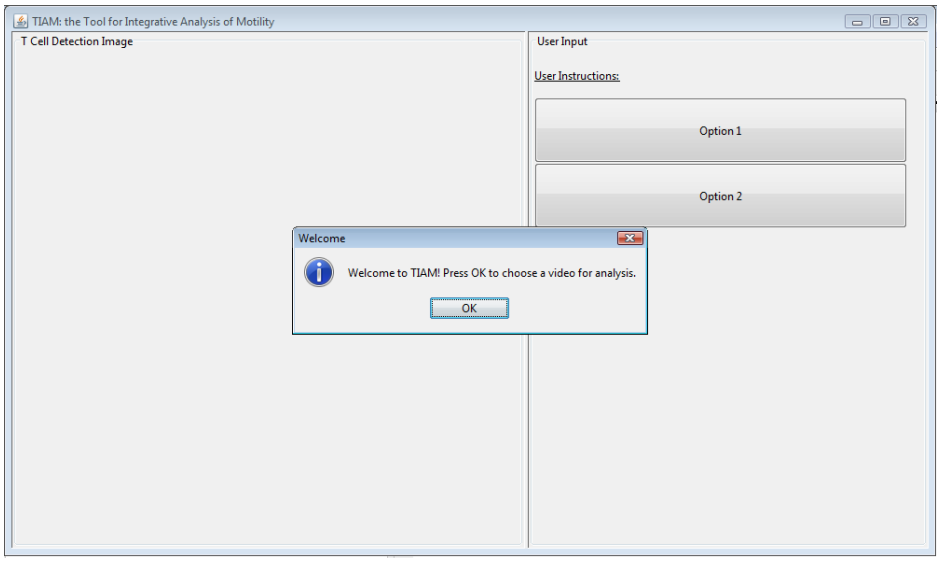
Download the TIAM github repository as a zip file. Extract and put it under the MATLAB directory for running TIAM.

The multi-channel image series needs to be provided in a folder as a set of image files in tif format wherein each file only contains information on one channel. The image files need to be named such that individual channel files are arranged in consistent order for every subsequent frame. For example: Demo_t01_c01.tif, Demo_t01_c02.tif, Demo_t02_c01.tif, Demo_t02_c02.tif and so on.

We use ImageJ software for this preprocessing as it has plug-ins to read data from most microscope vendor file formats. Menu option File->Save As->Image Sequence allows saving the data in the desired format for TIAM input.

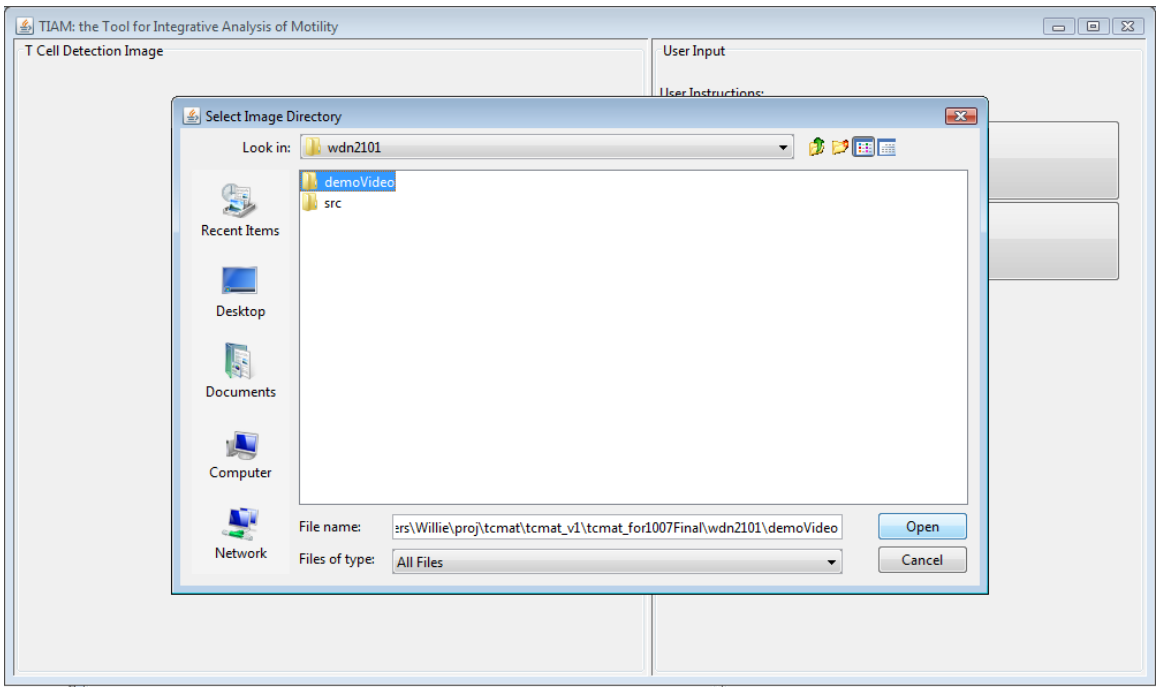
User Manual

To run TIAM, navigate to the TIAM/src directory in MATLAB, and type the name of the main function, tcmatMain, into the console. Once run, TIAM will display the following welcome screen.



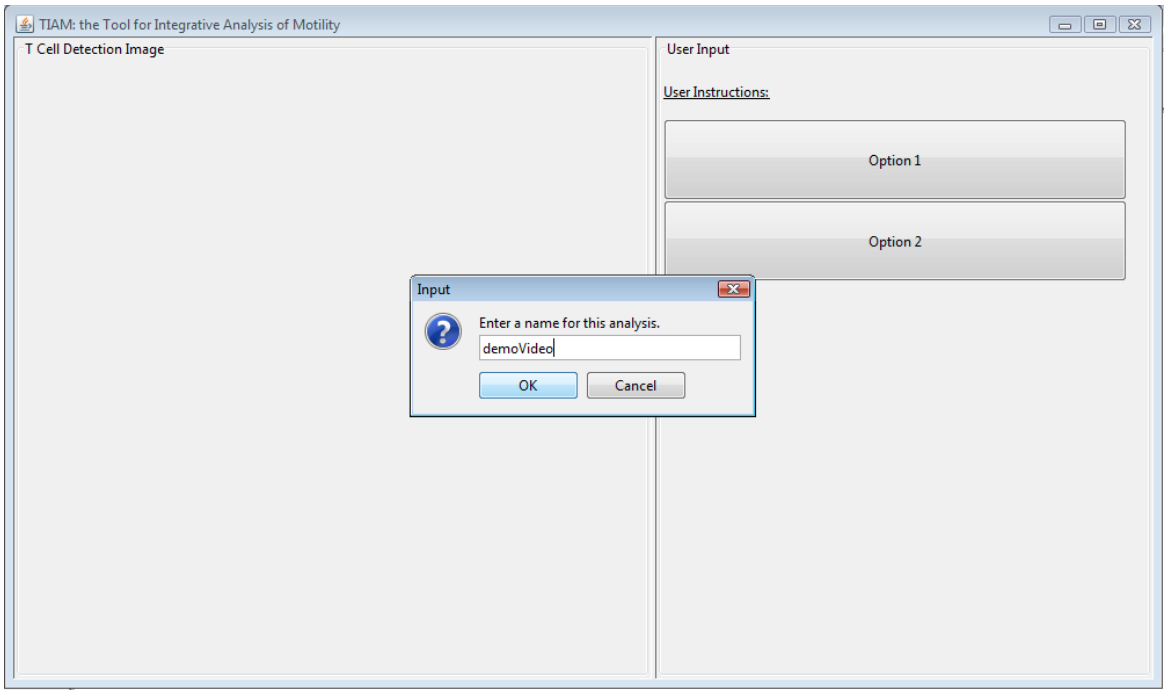
The user is then prompted to choose a “video folder”, which is a folder containing an image sequence from a multi-channel time lapse microscopy experiment.

To demo this tool, the user should choose the TIAM/doc/demoVideo folder, which contains images from a sample time lapse microscopy video of T cells (as shown below).



This demo image sequence has 19 frames with 4 channels. The first channel is reflection/IRM, 2nd is flur1, 3rd is flur 2(2nd fluorescence channel) and 4th is the DIC channel. The user must know the channel order a priori to ensure desired analysis by TIAM. In the current deployment, TIAM handles two fluorescence channels along with a DIC and a reflection channel.

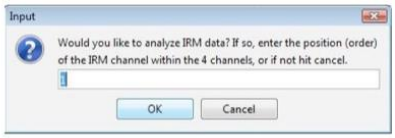
The user must then specify information about the video to be analyzed. This is necessary to provide information such as the number of different channels in the multi-channel time lapse microscopy video, and which features the user wants analyzed. To begin, the user is prompted to enter a name for the analysis as the figure below shows.

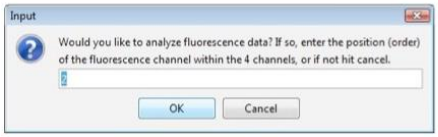


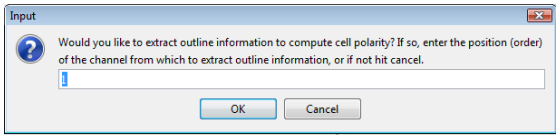
After entering a name, the user is prompted to enter information about the number of channels, the ordering of the channels in the image file sequence, and what information is desired from the analysis. In the demo video (i.e. for the images in demoVideo), the correct information that must be entered is:

- In name prompt: <any name is fine>
- In number of channels prompt: 4
- In DIC channel prompt: 4
- In IRM channel prompt: 1 (this is optional)
- In Fluorescence channel prompt: 2 (this is optional)
- In second Fluorescence channel prompt: 3 (this is optional)

As shown below:



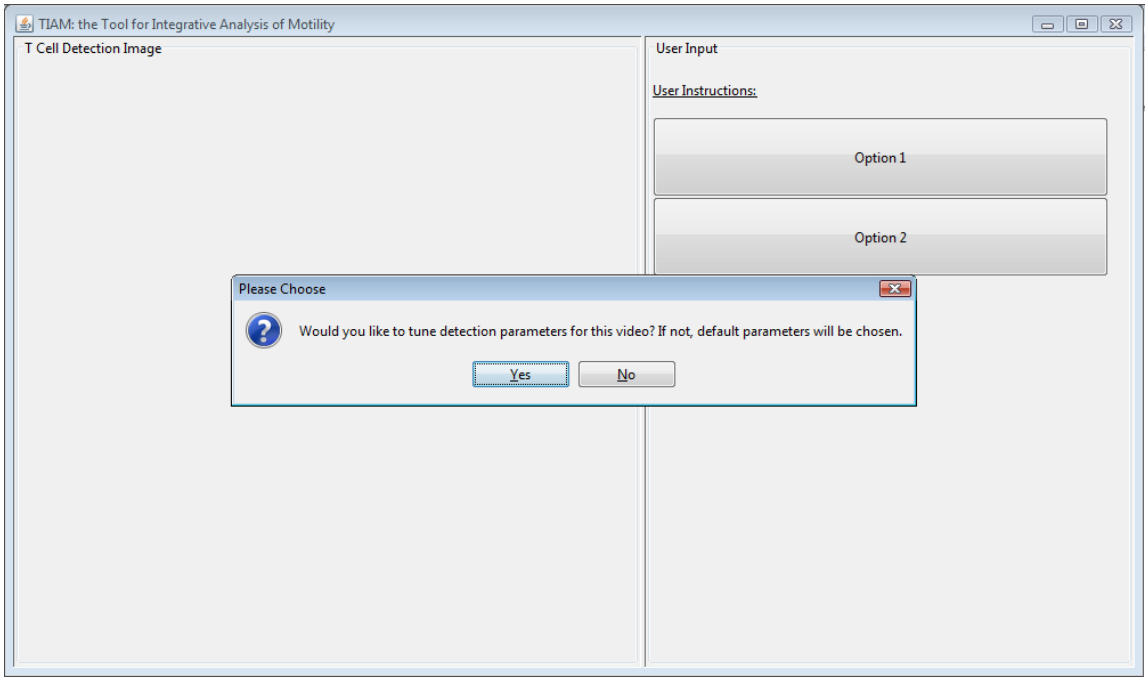




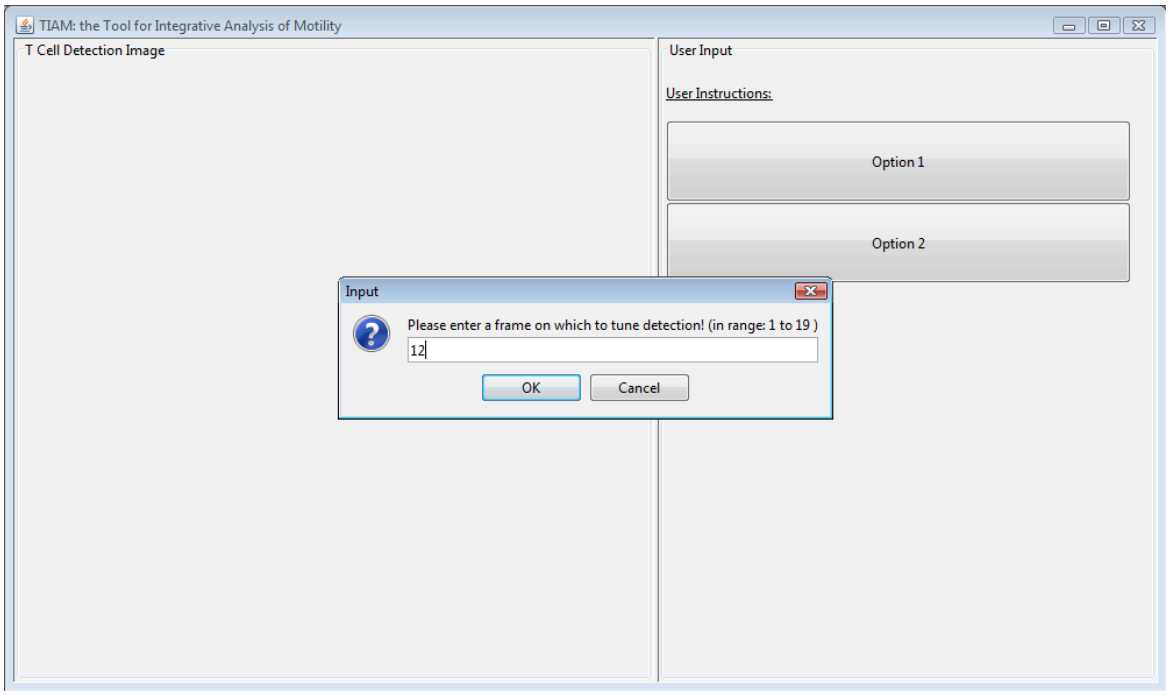
(note that the video information values for the demo video are given as default values, as our T cell video time lapse microscopy image sequences often contain this number and order of channels).

As instructed in the prompt, hitting ‘cancel’ is an option that the user may choose if certain channels are present in the image sequence but the user is not interested in extracting information from that channel.

Next, the user must choose whether to tune the parameters of the detection algorithm for the given video, or whether to use a set of default parameters, as shown in the image below



If the user selects no, default detection parameters are chosen. If the user selects yes, the user is prompted to enter a frame in the video to tune the detection parameters on, as shown in the following image:



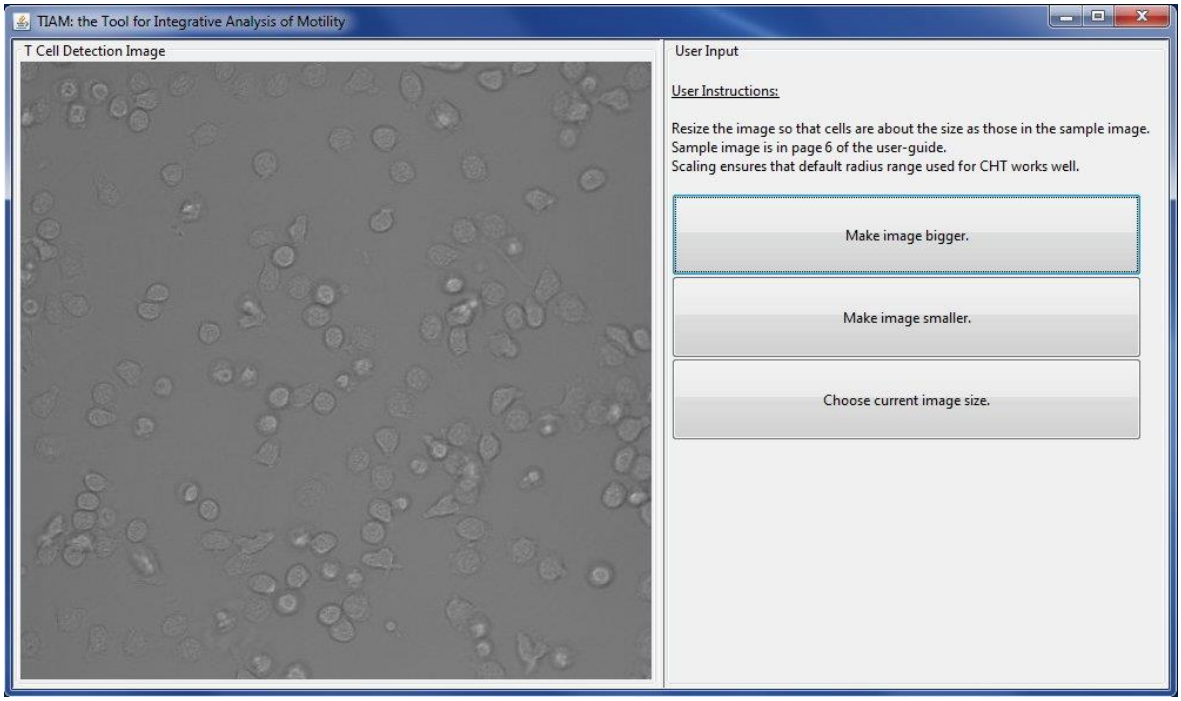
The user is then able to adjust the following parameters in a sequential manner:

- size of the image (which ensures that the default radius range of CHT work well)
- brightness of the image
- edge sensitivity (when taking edge filter of image)
- Hough transform array size (for fine tuning the radius range for CHT)
- cell search radius (for choosing maxima over imperfect circles)

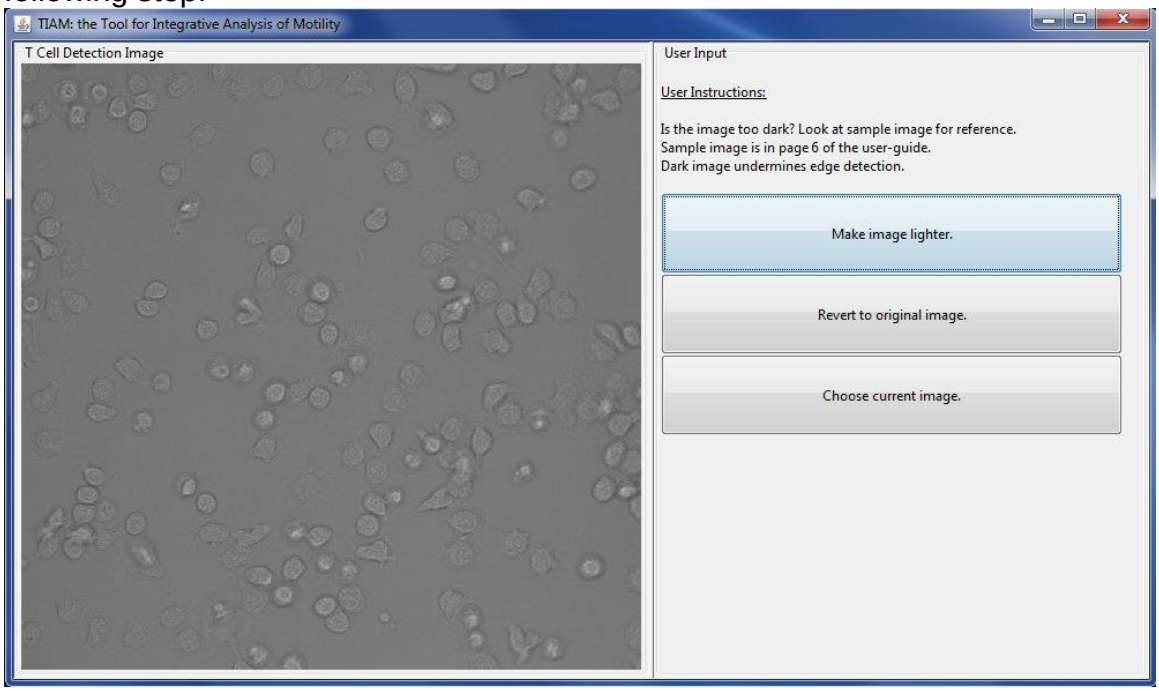
Each of the above parameters comes with a visualization that allows the user to see the effect of changing the parameters. Choosing the correct parameters is a learned process that a user can get a feel for over time. Processing the demo video should help in this process. The user should also be aware that it is not possible to come back to the previous step in the user-interface. Also, it is not possible to abort the user-interface, until the TIAM analysis starts.

The following images show good detection parameter settings for the cells in the demo video. The user should try to choose their detection settings such that the displayed images look as similar as possible to the following images for each of the steps.

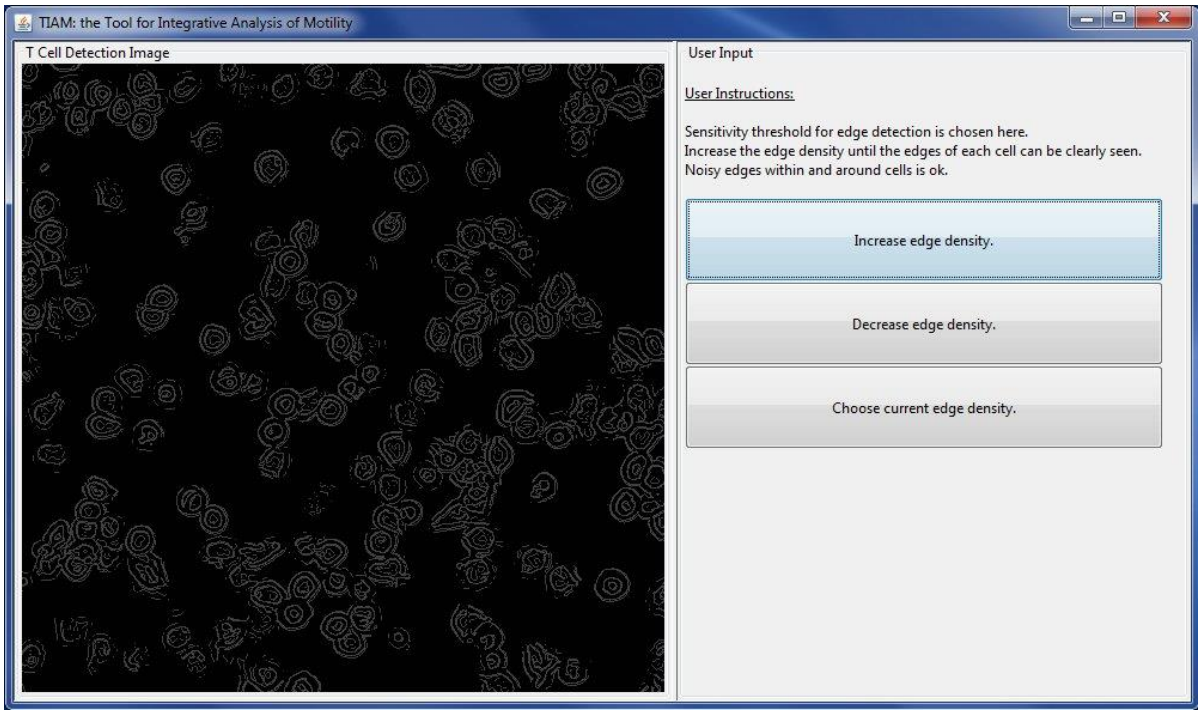
The following image shows a good size/scale parameter setting:



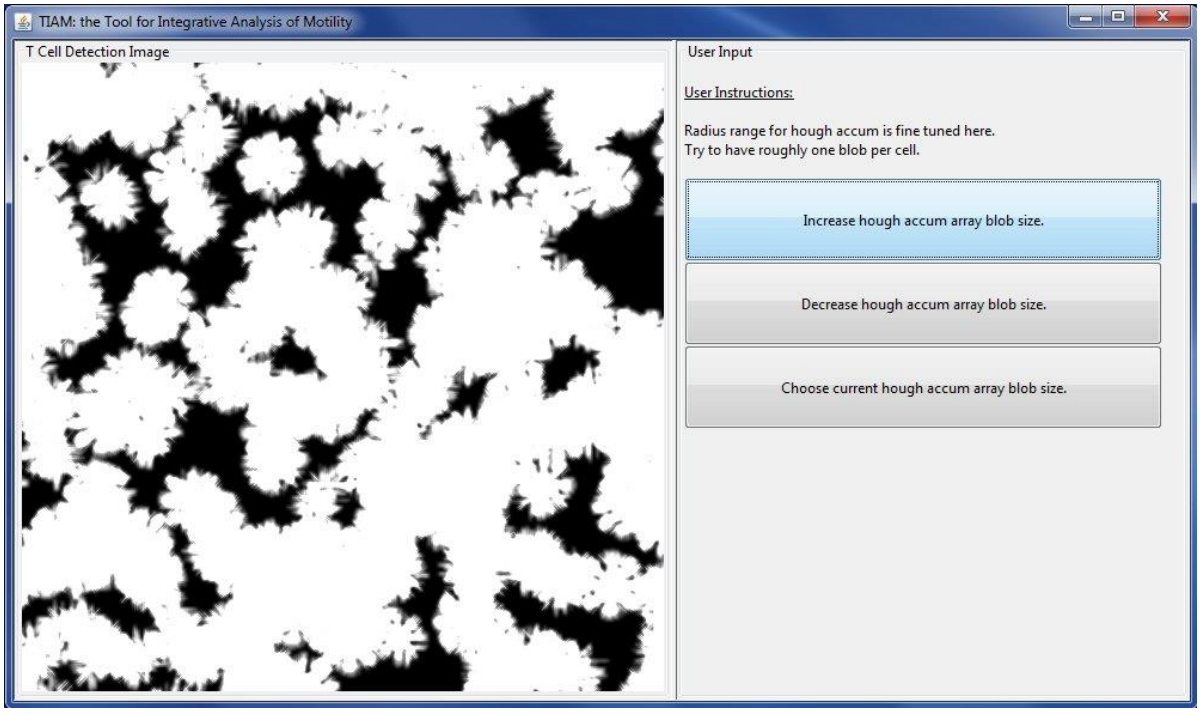
The transimitted light images can sometimes be too dark with very little contrast. Such images can be transformed to accentuate the edges, if desired in the following step.



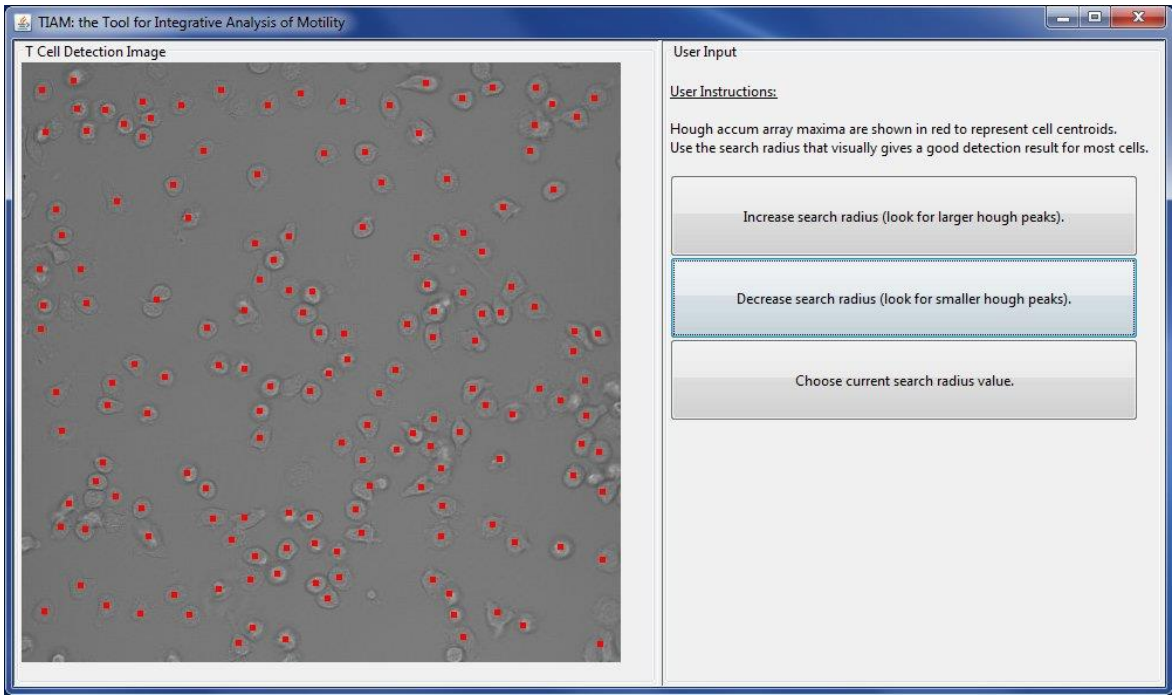
The following image shows a good edge sensitivity parameter setting:



The following image shows a good Hough transform radius range parameter setting

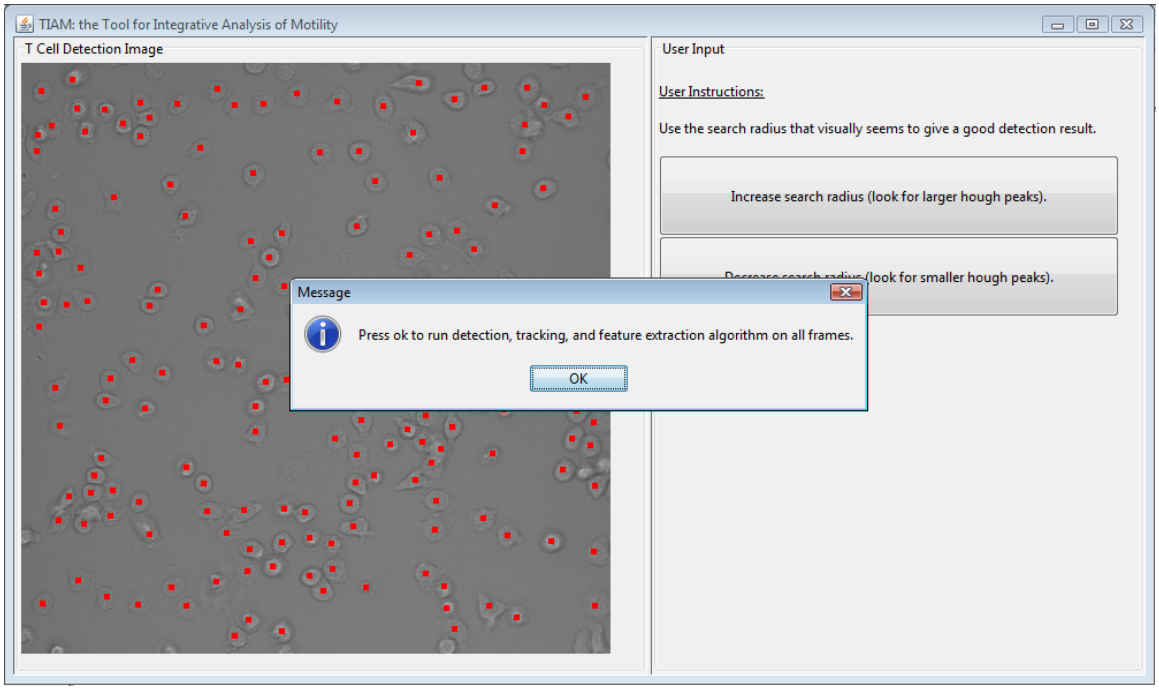


The following image shows a good cell search radius parameter setting:

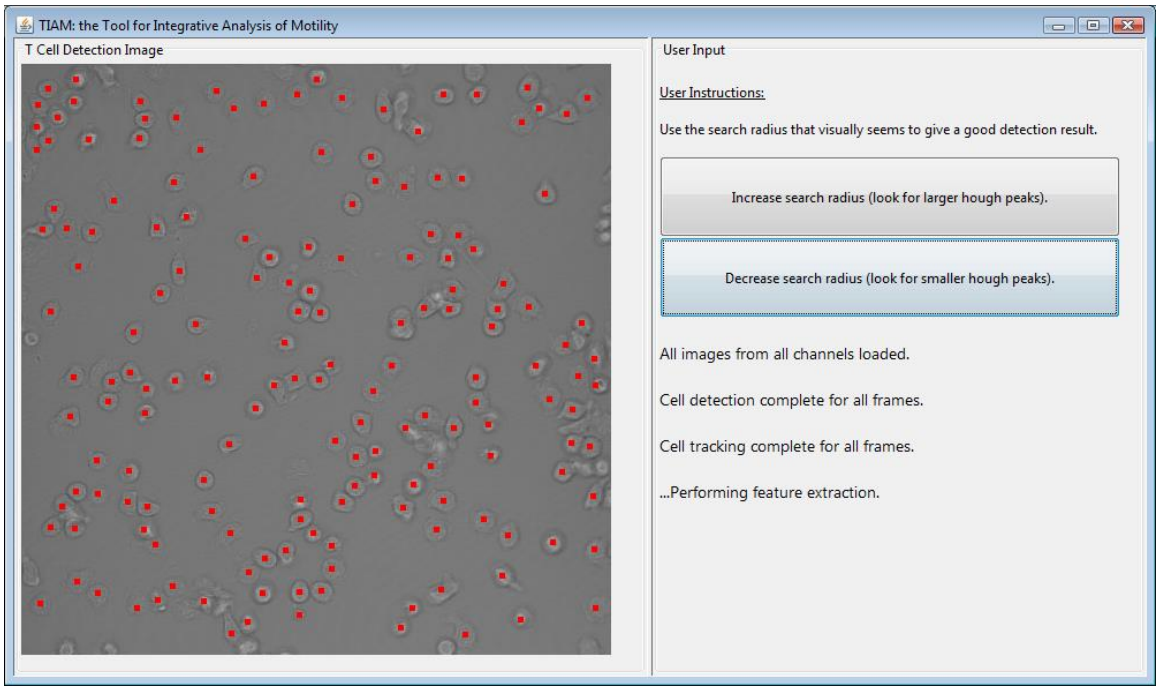


The results of detection for this frame are shown in this final image.

After the user selects the final parameter, he or she is prompted to run the entire analysis algorithm (detection, tracking, and feature extraction) on all frames in the video, as seen in the figure below:



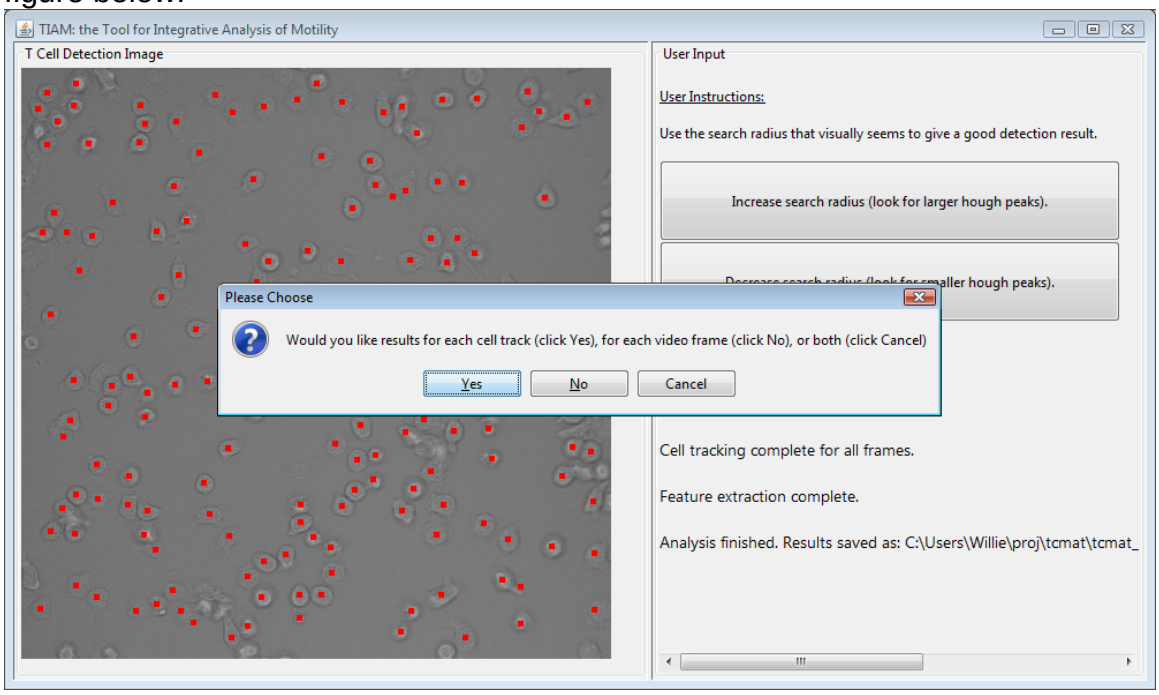
After user confirmation, detection for each frame will proceed, results of which are displayed on the left-hand side. Tracking and feature extraction steps will ensue. As the algorithm progresses, each step it carries out is logged on the lower right hand portion of the GUI, as seen in the figure below:



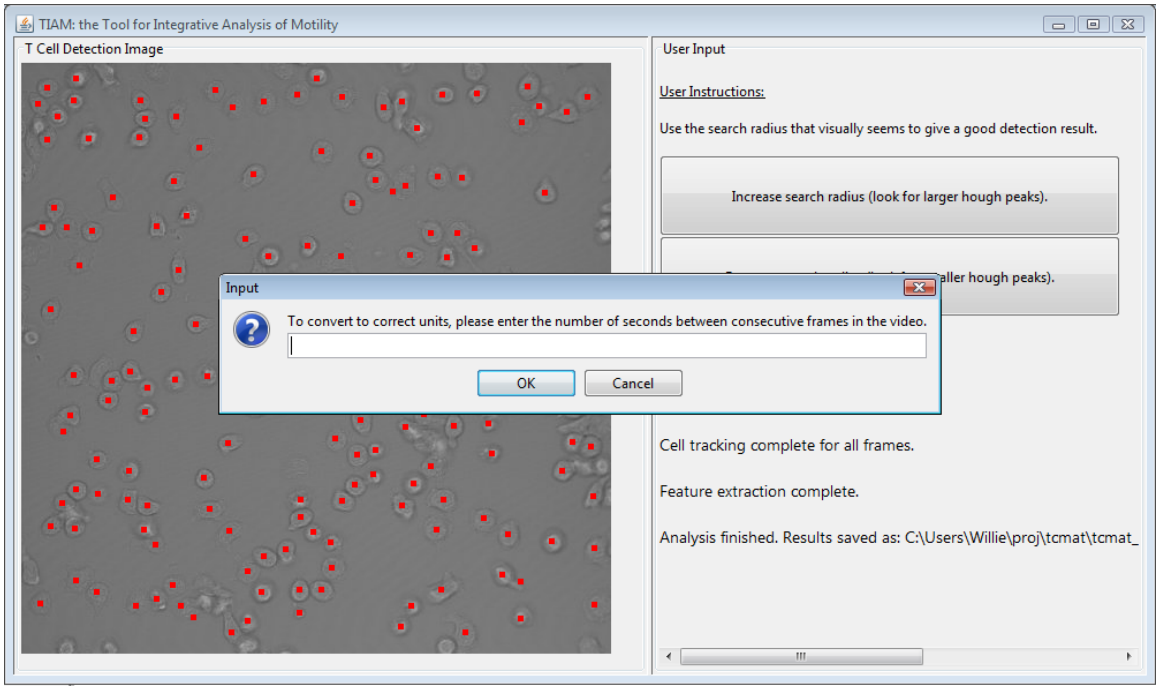
The analysis as a whole can take varying lengths of time, mostly dependent on the size of each frame, the number of cells in the experiment, the number of

frames in the video, and whether feature extraction steps are included in the analysis.

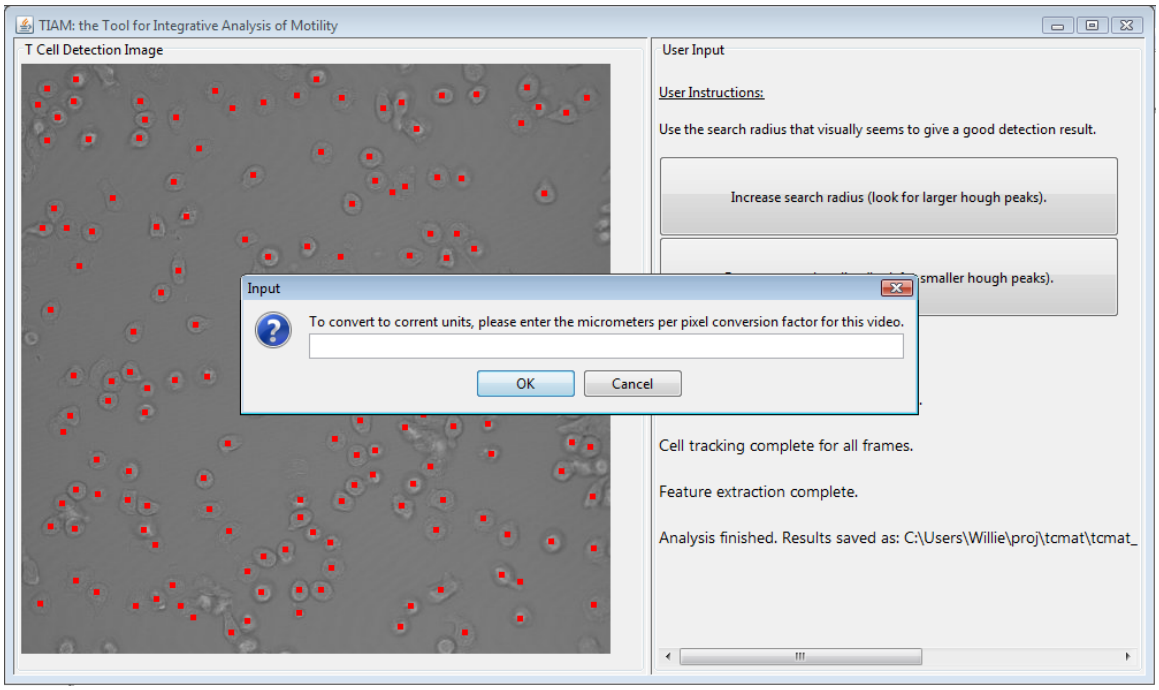
The user is then given the choice of getting motility analysis results for each cell track, for each video frame (averaged over all cells), or both, as shown in the figure below:



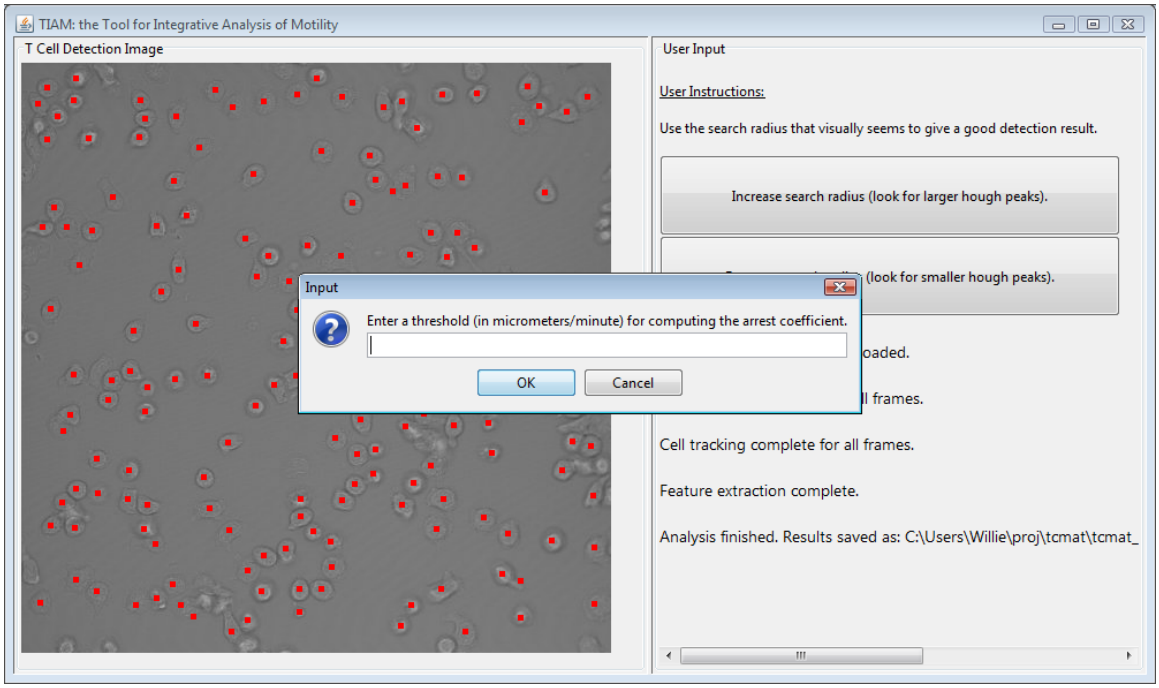
The user is then prompted to enter the number of seconds between frames in the time lapse microscopy video. This allows for speeds and other motility features to be correctly calculated. Entering a value of 30 here suffices for the demoVideo.



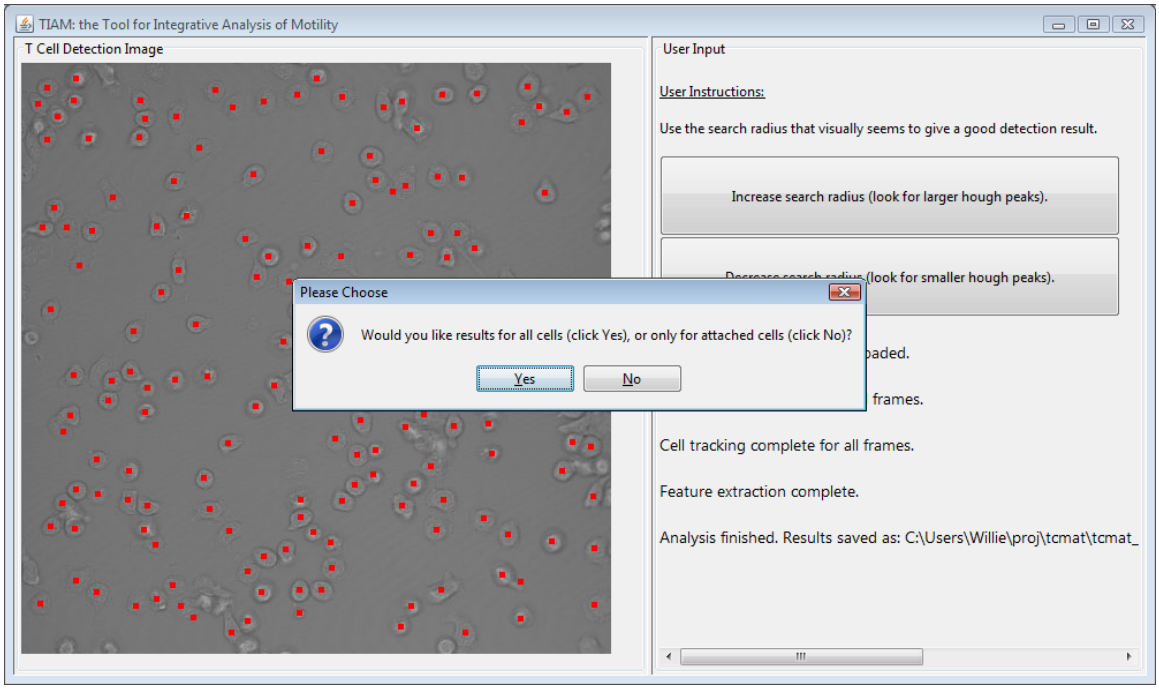
The user is then prompted to enter the size-conversion factor (the number of micrometers per pixel) for each of the time lapse microscopy images. This allows for speeds, cell areas, and other motility features to be correctly calculated. Entering a value of 0.439 here suffices for the demoVideo.



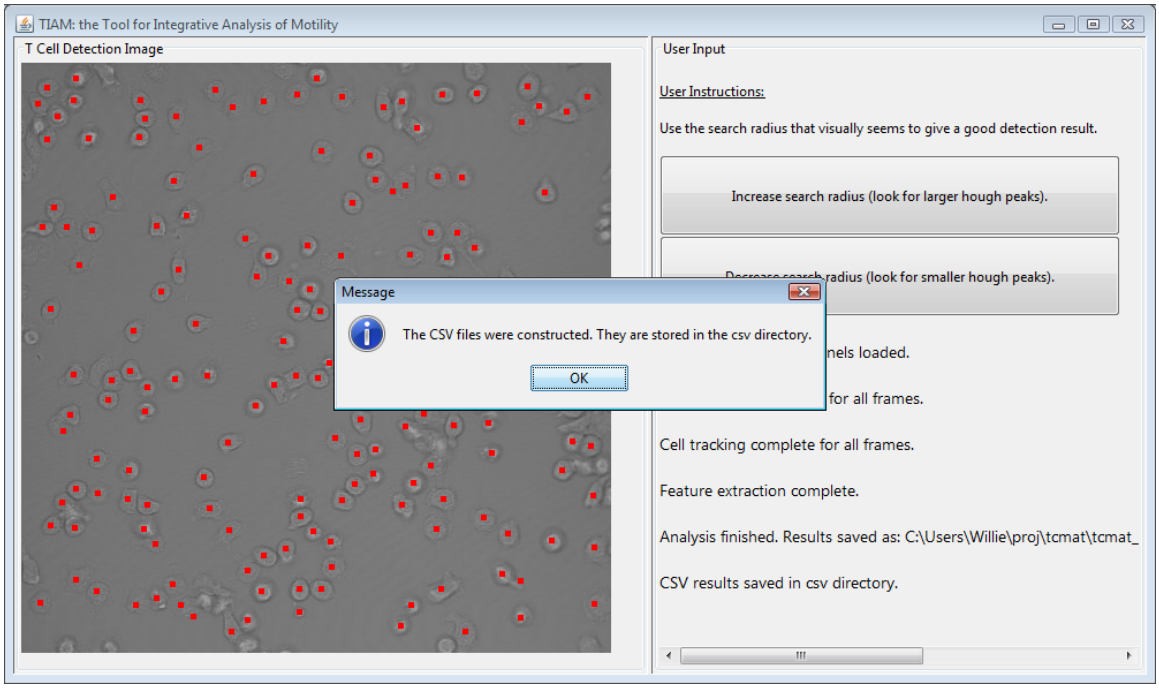
The user is then prompted to enter a threshold for computation of the arrest coefficient (a useful motility feature). Entering a value of 1.0 here suffices for the demoVideo.



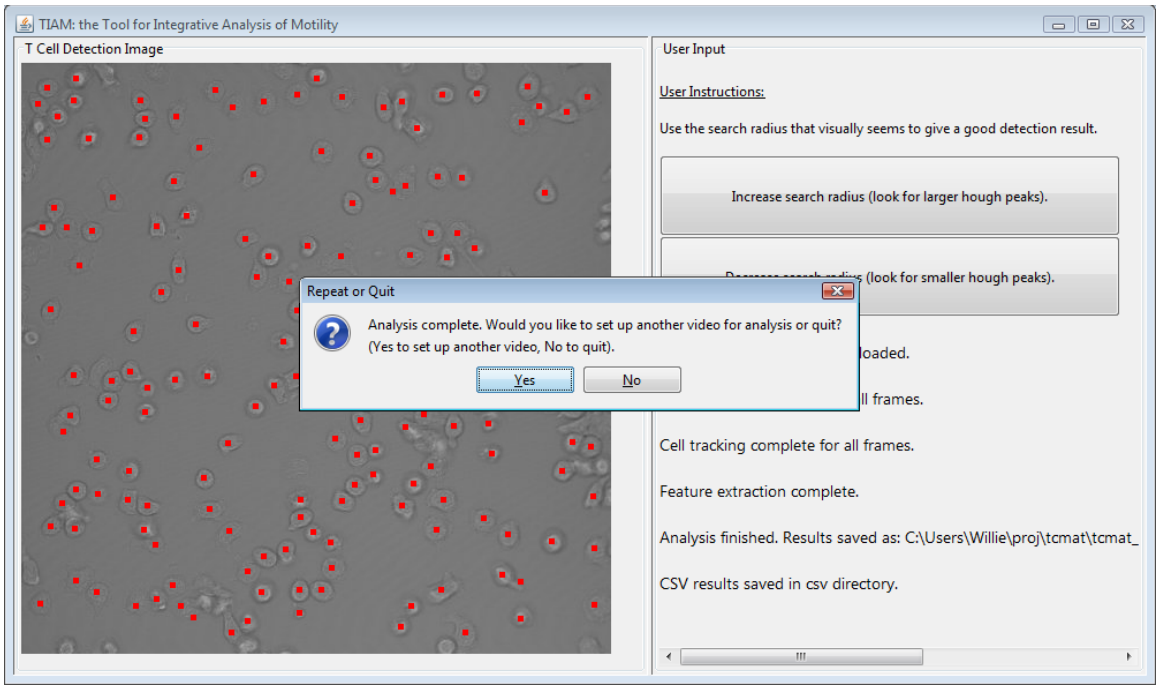
The user can the choose whether to extract data for each cell, or only cells that have attached to some underlying substrate (i.e. cells whose IRM attachment area is non-zero).



After getting all the necessary information, TIAM computes the motility results and saves them in CSV format. The CSV files are placed in the TIAM/csv directory with the name <name of analysis>_perCellTrack.csv or <name of analysis>_perFrame.csv



Once the full analysis is complete, the user will be notified via a popup box. Additionally, results of the analysis will be saved in an output file in the TIAM/ws folder (as <name of analysis>_results.mat), in a .mat MATLAB formatted binary file, which can be re-opened in MATLAB for later exploration. If a channel had been chosen for extracting outlines, a tiff series file with the cell outlines will be saved in the TIAM/ws folder with the name <name of analysis>_outline.tif. The user is also prompted to choose whether to carry out analysis on another video or exit the program (if the user chooses to exit, the final state of the program is frozen, and the user is able to view all displayed log information until he or she exits the program by closing the window). This is shown in the figure below:



Format of the *.mat output file:

Note that in the resulting MATLAB file (saved in the ws/ folder), the primary data structure containing the results is called “datacell”. The datacell is a 1-dimensional MATLAB cell array, where each element is a matrix that holds data relevant to a single cell-track (and each row in a given matrix corresponds to data from a single time-step). Each cell-track-matrix has the following motility fields in the given columns:

1: startframe *
2: endframe *
3: x position
4: y position
5: irm attachment area
6: fluor channel 1 value
7: fluor channel 2 value
8: fluor / cell-type classification *
9: step 1 speed
10: step 4 speed
11: step 8 speed
12: normalized displacement *

13: displacement *
14: arrest coefficient *
15: turn angle
16: corrected confinement index *
17: eccentricity (as a measure of morphological polarity)
18: circularity (as a measure of morphological polarity)
19: aspect ratio (as a measure of morphological polarity)

* asterisk means the value is the same for all rows

Format of output .csv files:

In the output .csv files titled <name of analysis>_perCellTrack.csv, each row corresponds with a single cell track, and the motility fields in each column are:

1. Cell track index
2. Cell type (assessed from fluorescence channel)
3. Average 1-step smoothed speed
4. Average 4-step smoothed speed
5. Average 8-step smoothed speed
6. Normalized displacement
7. Displacement
8. Average IRM channel attachment area
9. Average first fluorescence channel value
10. Average second fluorescence channel value
11. Arrest coefficient
12. Average unsigned turn angle
13. Average signed turn angle
14. Confinement index
15. Average polarity (default is aspect ratio)

Note that averages in the above fields are taken over each frame for a given cell track.

In the output .csv files titled <name of analysis>_perFrame.csv, each row corresponds with a single cell track, and the motility fields in each column are:

1. Frame index
2. Average 1-step smoothed speed
3. Average 4-step smoothed speed
4. Average 8-step smoothed speed
5. Average IRM channel attachment area
6. Average polarity

Note that the averages in the above fields are taken over each cell in a given frame.

The *.csv files are record the detection parameters used in the analysis.

Batch script for analyzing multiple experiments

There are two ways running TIAM in batch mode.

One way is by running the tcmatBatchMain function under TIAM/src. For this the user needs to update the parameters in batchSetup.m under TIAM/src/matlab/. The currently available batchsetup.m and batchSetupExample.m under TIAM/src/matlab/ can be used as templates.

The second way is to use a batchSubmit.m file from any location with all the parameter values. This makes calls to TIAM functions. The currently available batchSubmit_052713_example.m and batchSubmit_exp1_rVary.m under TIAM/src/matlab/ can be used as templates.

Default detection parameter values:
We typically use, for a 40x lens with NA 1.3, the following set of parameters for detecting primary T cells. These are also default values under user-interface based mode of TIAM:

imageScale = 1.2
edgeValue = 0.1
radiusMin = 5
radiusMax = 15
gradientThresh = 10
searchRadius = 15
minCellSeparation = 5
darkImage = 0

Important parameters for advanced users and potential TIAM developers

Those wishing to modify TIAM should take note of important (fixed) parameters in TIAM's algorithms whose values may affect TIAM's performance. The following list gives the parameter name, the function where it resides, the line number within that function, and the corresponding variable name in that function:

Link-length for nearest neighborhood
    File: tcmatAnalyzeVideo.m
    Line: 364
    Variable: max_trackingjump
Bounding box area for feature extraction

File: tcmatAnalyzeVideo.m
Line: 393
Variable: halfCropSize

The same parameters are also present in tcmatBatchScript.m.

Choice of polarity measure
File: makeResultCsvs
Line: 59, 114

Analogous positions in makeResultCsvs_batch.m

Threshold for segment joining
File: joinSubtracks_new2.m
Line: 106
Variable: joinThresh

The morphological processing parameters under the feature extraction functions may also be changed. The functions are named as such.

Editing Tracks with TIAM

TIAM's track editing function, editTracks.m, allow a user to manually edit tracks. This function has input arguments:

- datacell
- breaksFileString
- joinsFileString
- arrestCoefThresh
- arrestCoefDatacellColumn

where:

- datacell is the data-structure output returned by TIAM.
- breaksFileString is a string showing the path to a file that encodes how tracks should be split into segments. Each line in the 'breaksFile' corresponds with a single track, and has the form:
  <trackIndex> <endFrameOfFirstSegment> <endFrameOfSecondSegment> ...
  where *trackIndex* is the index of the track in the datacell. An example line is given: 25 35 78 150 (This creates 4 segments of the track '25': 1-35; 36-78; 79-150; 151-last frame)
  Note that all tracks that will be used for joining will have to be in the 'breaksFile'. Even the tracks that actually won't be broken will have to be included with the *<endFrameOfFirstSegment>* denoting the last frame of the track.
- joinsFileString is a string showing the path to a file that encodes how broken track segments should be joined together. Each line in the 'joinsFile' corresponds to a single resulting track in the output, and has the form:
  <firstTrackIndex> <firstSegmentIndex> <secondTrackIndex> <secondSegmentIndex> ...
  where *trackIndex* is the index of the track in the datacell and *segmentIndex* specifies which segment to use out of those created by 'breaksFile'. An example line is given: 25 1 32 2 (This joins 1st segment of 25th track to 2nd segment of 32nd track to give a new edited track).
- arrestCoefThresh is the arrest coefficient threshold given by a TIAM user.
- arrestCoefDatacellColumn is the column of the datacell on which the arrest coefficient is calculated, which is also specified by a TIAM user.

The content of the 'breaksFile' and the 'joinsFile' are decided based on visualizing the tracks to look for errors in tracking. Instruction for visualizing tracks is explained next.

Dynamic visualization of tracks

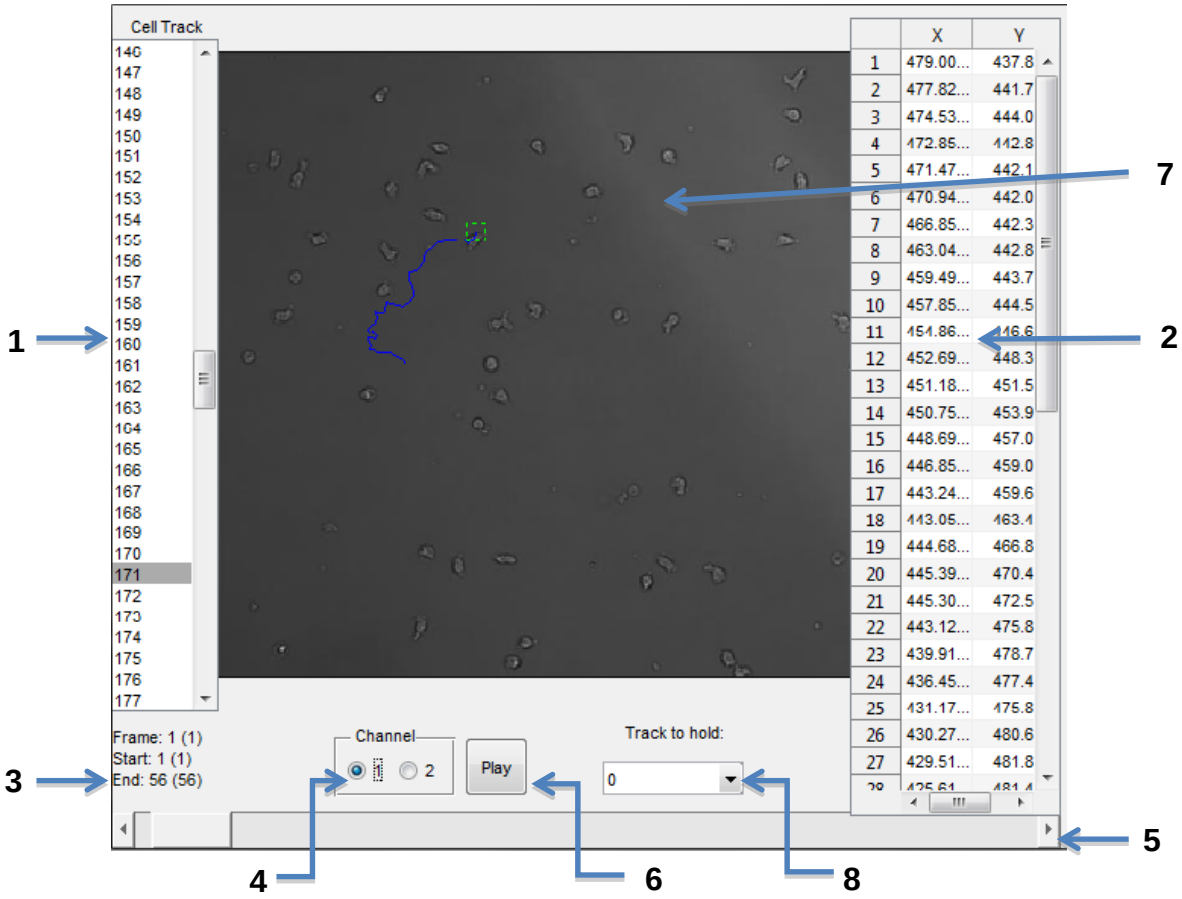
Code for dynamic visualization of tracks is stored in TIAM/src/trackGUI/

The function joinTracks is used for dynamic visualization of tracks. This function needs to be in a directory that also contains the 'data' folder with the results from TIAM. To run joinTracks, the following variables (in the trackGUI_OpeningFcn function) have to be set by the user:

- **tiamFile** – this should begin with “/data/” and the contain the path to an output file of TIAM, which is a .mat cell array
- **ch1PicNames** – and **ch2PicNames** are set by dir(fullfile()). The fullfile() function takes as arguments the parts of the paths to files. The arguments should be of the form “pwd,’data’,[some directory], ‘*.jpg’.” The “*.jpg” expression are supposed to be different for **ch1PicNames** and **ch2PicNames** as they refer to the image-series corresponding to different channels. The paths of image series need to be specified again for the imread() function to load the images onto MATLAB.
- **handles.frameSizeX** – this should be the size in µm of the horizontal dimension of the images.
- **handles.frameSizeY** – this should be the size in µm of the vertical dimension of the images.

The user may choose to only display a portion of the image-field by changing the axis dimensions in line 391.

Once these variables are properly set, navigate to [the directory containing trackGUI.m] in MatLab and type trackGUI in the command prompt. The following should display (see next page):



The features of the user-interface are explained below:

1. Contains a list of all the cell tracks in the tiamFile. When the user runs trackGUI, the first track is selected but the user may click any of the tracks and can scroll through the list of tracks. After a track is selected the main window (**7**) displays the frame at which that track starts. The curve of that track is drawn in blue. A lime-green bounding box of a pre-set size is also drawn.
2. Diplays the x and y position of the selected track indexed relative to the frame at which that track starts.
3. Provides information about which frame is being displayed as well as at which frames the selected cell track starts and ends. The parenthetical numbers correspond to the indexing in **2**.
4. Lets the user pick a channel.
5. The scroll bar's relative position corresponds to the current frame being displayed. The user can move it around or click the arrows at either end to navigate to other frames. When the play button(**6**) is down, **5** scrolls automatically.
6. When pressed, the frames are cycled through at a constant rate, thus displaying a video. When the the end frame of the track is reached the video stops playing. One may also stop at any desired frame by clicking on the play

button (**6**) as the video is playing.

7. Displays the frames and the tracks.
8. Allows the user to select a track to be held on the screen (in red) regardless of which cell track is selected in **1**. This facilitates detection of track swapping and also potential tracks that correspond to the same cell.