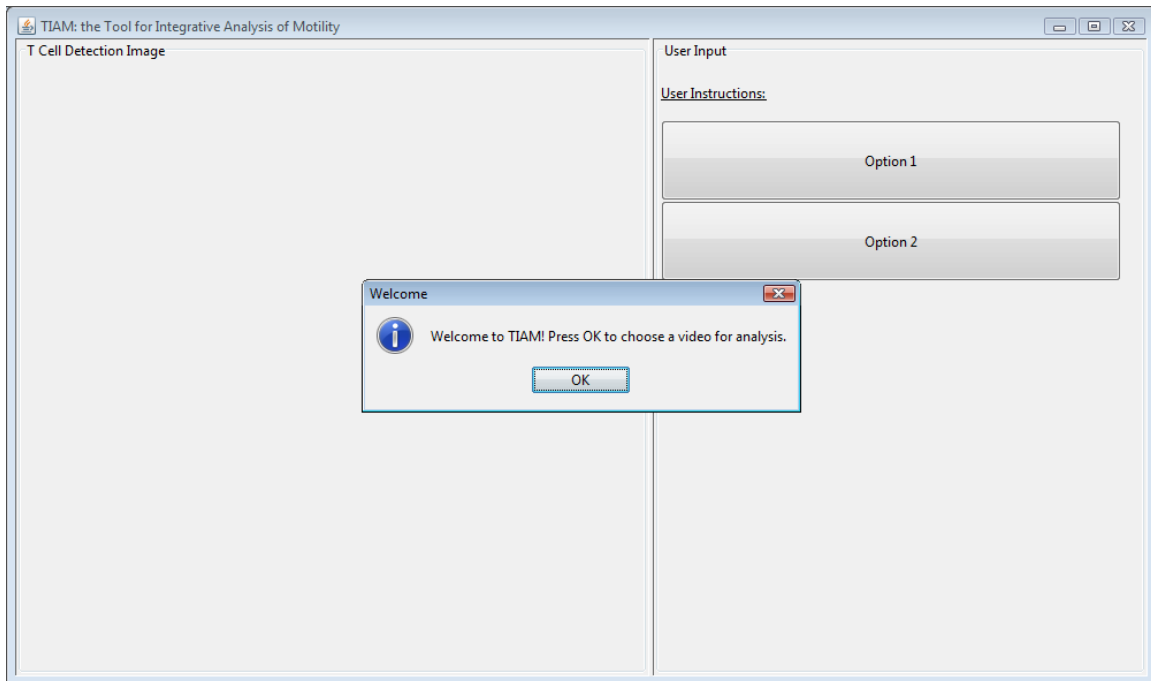


# TIAM: the Tool for Integrative Analysis of Motility

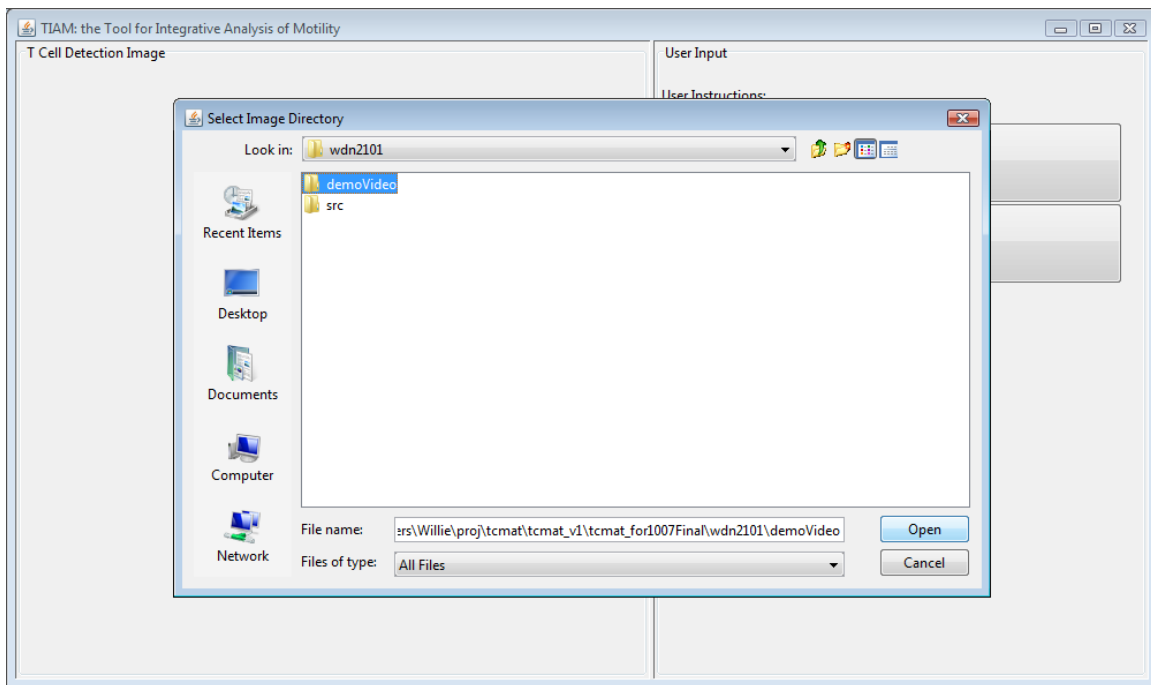
## User Manual

To run TIAM, navigate to the TIAM/TIAM\_v01/srcdirectory in MATLAB, and type the name of the main function, tcmatMain, into the console. Once run, TIAM will display the following welcome screen.

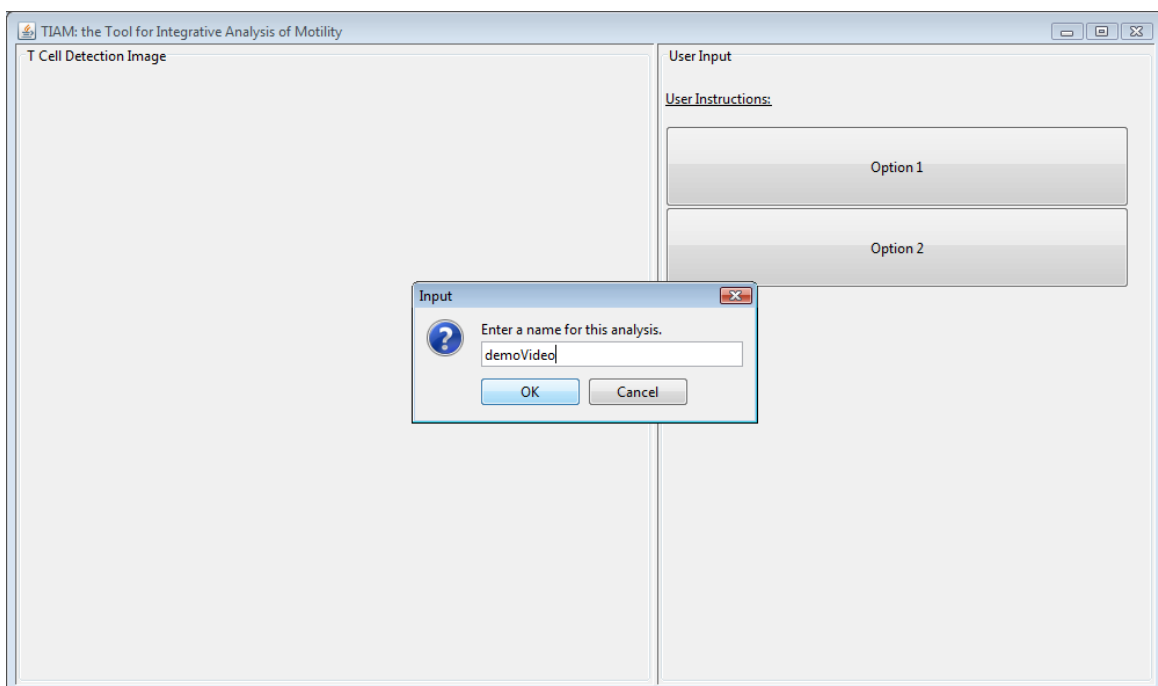


The user is then prompted to choose a “video folder”, which is a folder containing an image sequence from a multi-channel time lapse microscopy experiment. The images in this folder should be in a MATLAB readable format (e.g. .jpg, .tif, etc.).

To demo this tool, the user should choose the TIAM/TIAM\_v01/doc/demoVideo folder, which contains images from a sample time lapse microscopy video of T cells (as shown below).



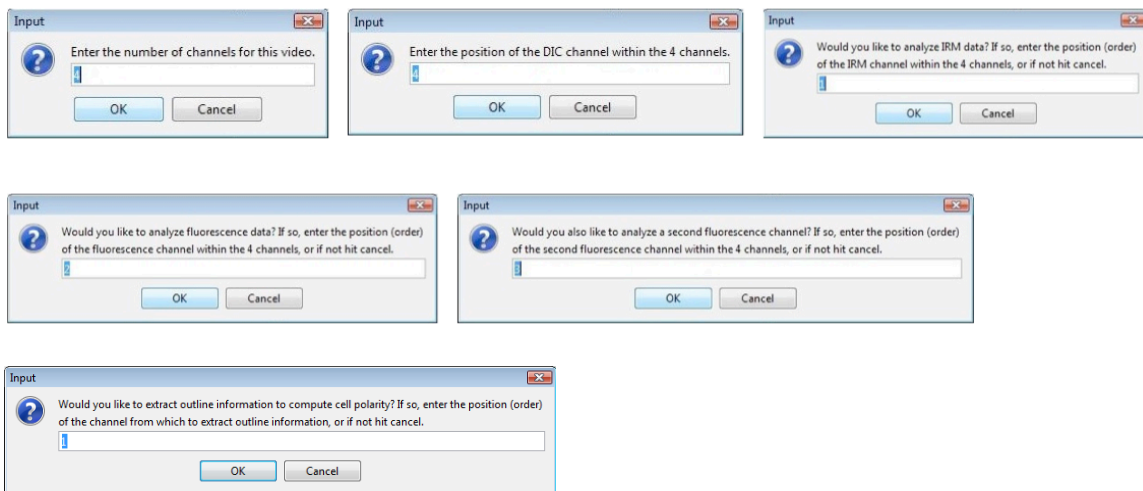
The user must then specify information about the video to be analyzed. This is necessary to provide information such as the number of different channels in the multi-channel time lapse microscopy video, and which features the user wants analyzed. To begin, the user is prompted to enter a name for the 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, and what information is desired in the analysis. In the demo video (i.e. for the images in demoVideo), the correct information that must be entered is:

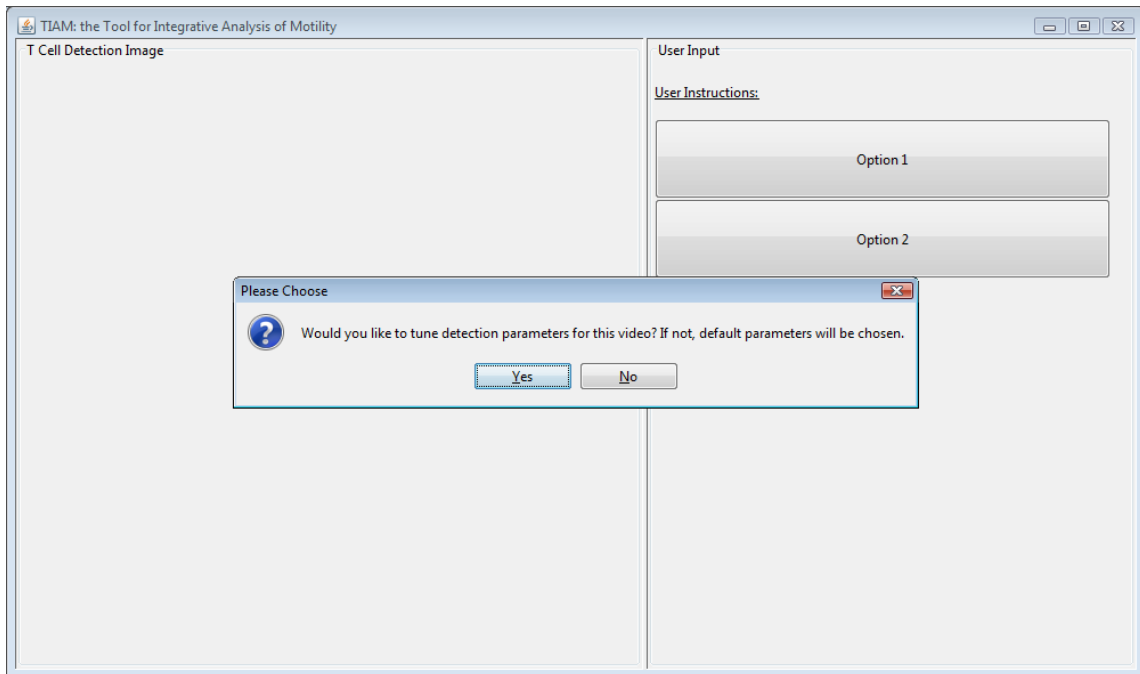
- In name prompt: <any name is fine>
- In number of channels prompt: 4
- In DIC channel prompt: 4
- In IRM channel prompt: 1 (this is optional)
- In Fluorescence channel prompt: 2 (this is optional)
- In second Fluorescence channel prompt: 3 (this is optional)

As shown below:

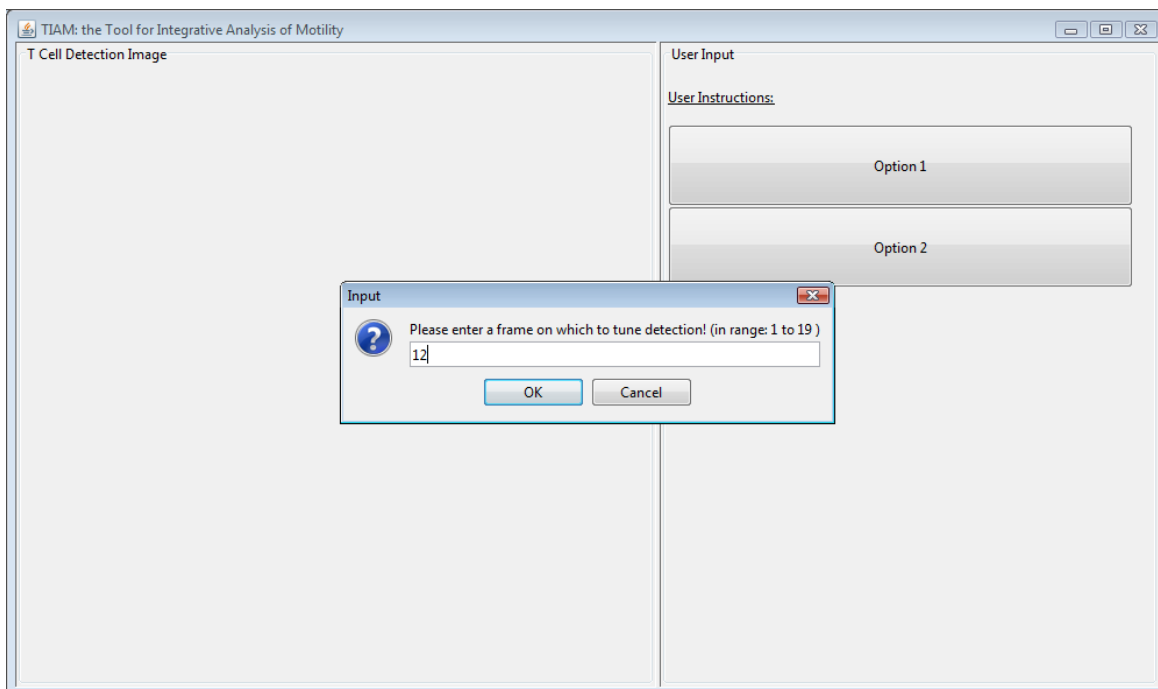


(note that the video information values for the demo video are given as default values, as T cell video time lapse microscopy image sequences often contain this number and order of channels).

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:



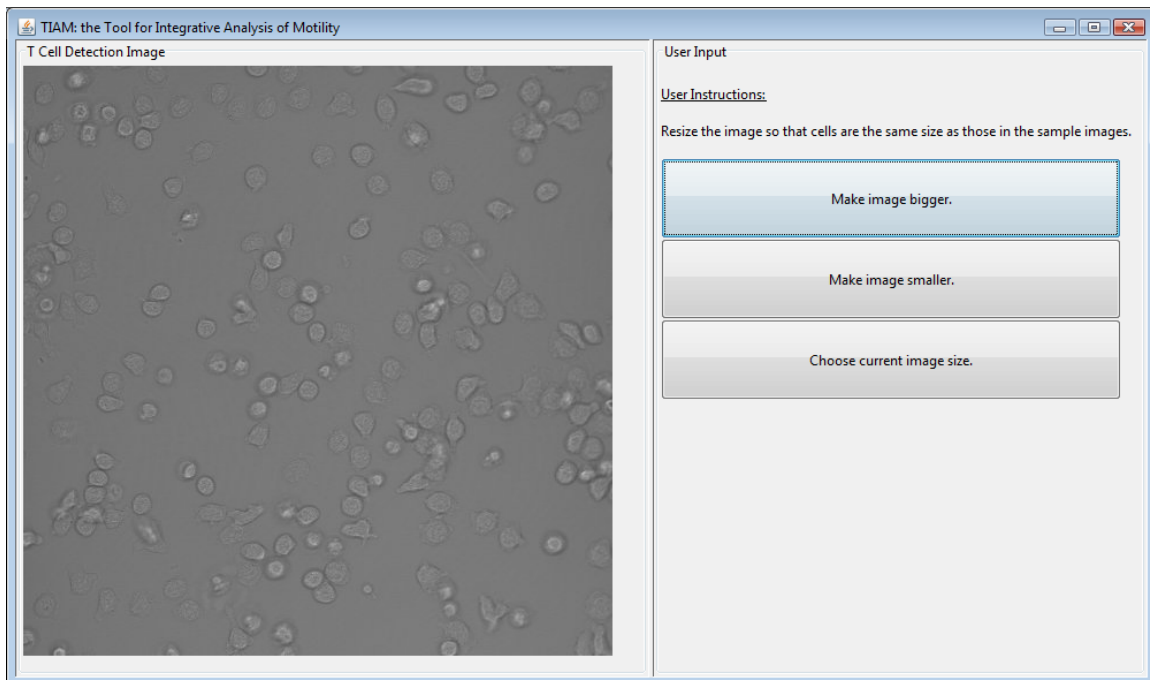
The user is then able to adjust parameters controlling the:

- size of the image
- brightness of the image
- edge sensitivity (when taking edge filter of image)
- Hough transform sensitivity
- cell search radius

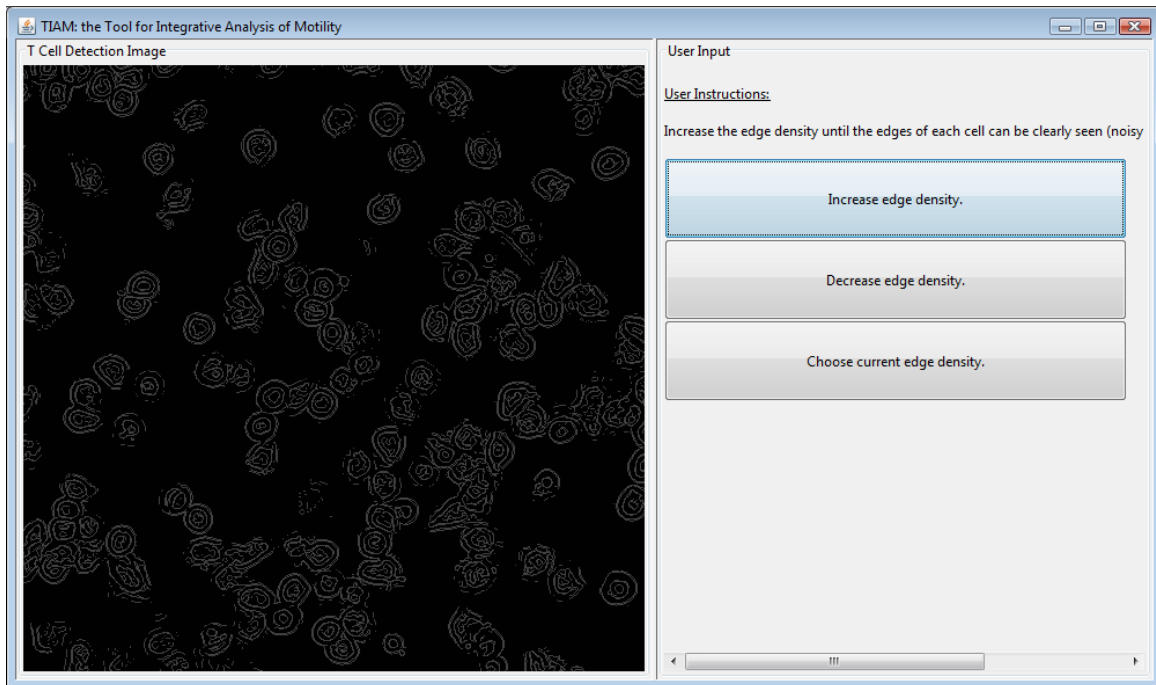
Each of the above parameters comes with a visualization that allows the user to see the effect of changing the parameters. Choosing the correct parameters is a learned process that a user can get a feel for over time.

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

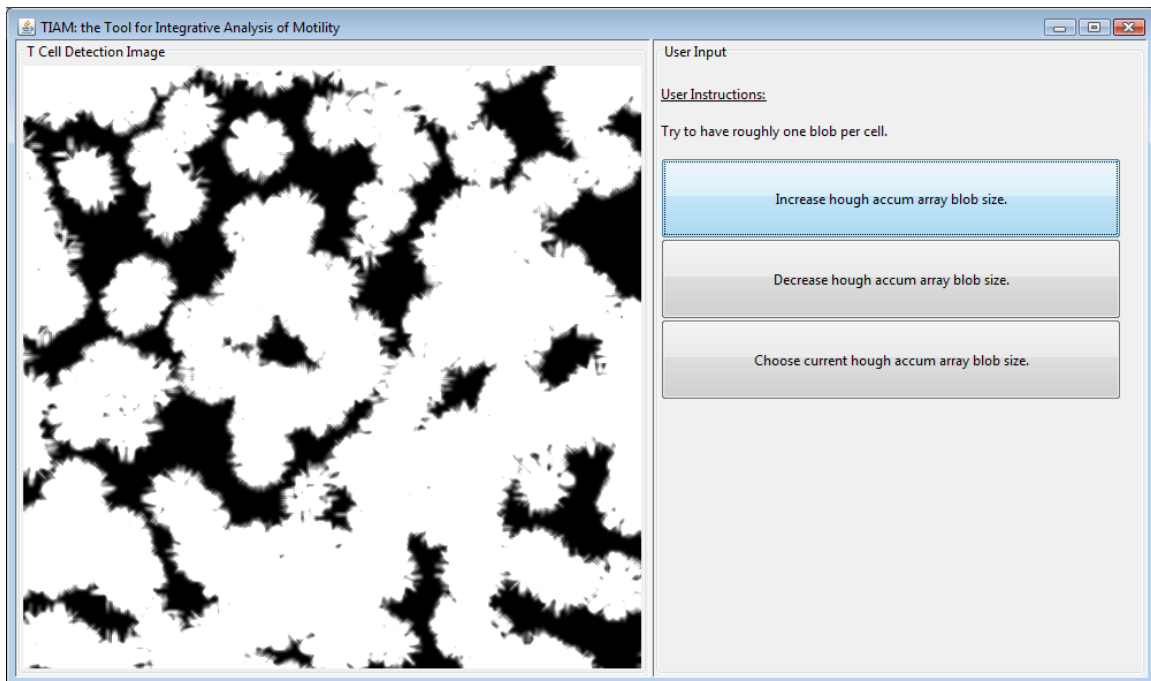
The following image shows a good size parameter setting:



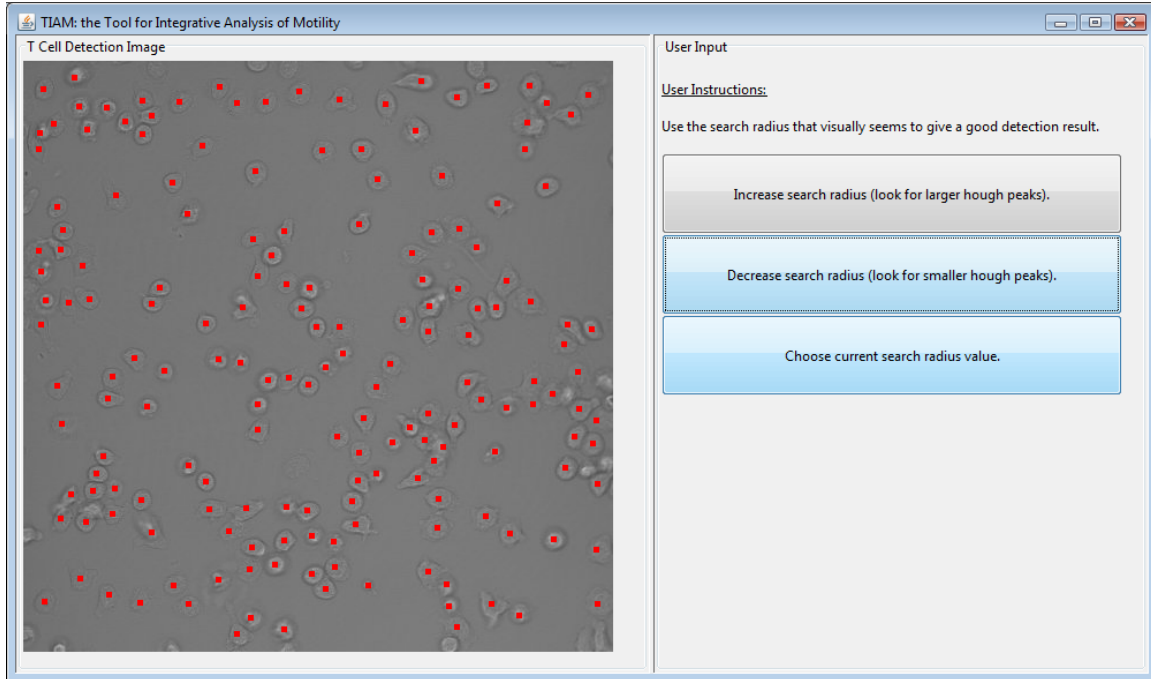
The following image shows a good edge sensitivity parameter setting:



The following image shows a good Hough transform sensitivity parameter setting

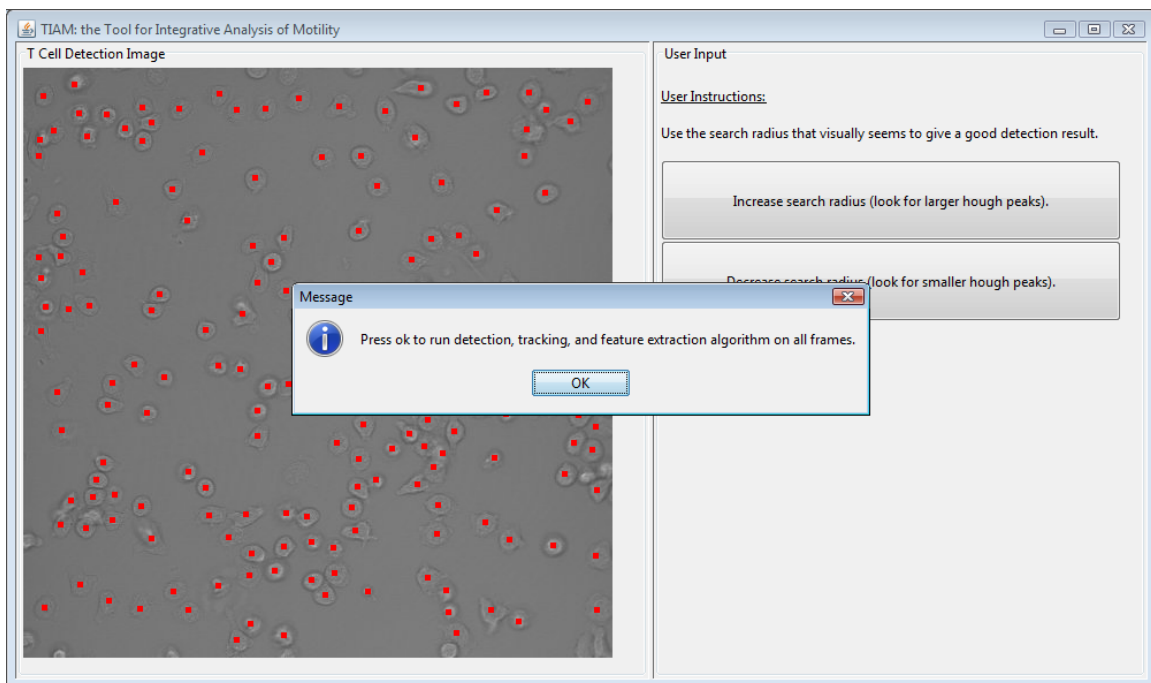


The following image shows a good cell search radius parameter setting:



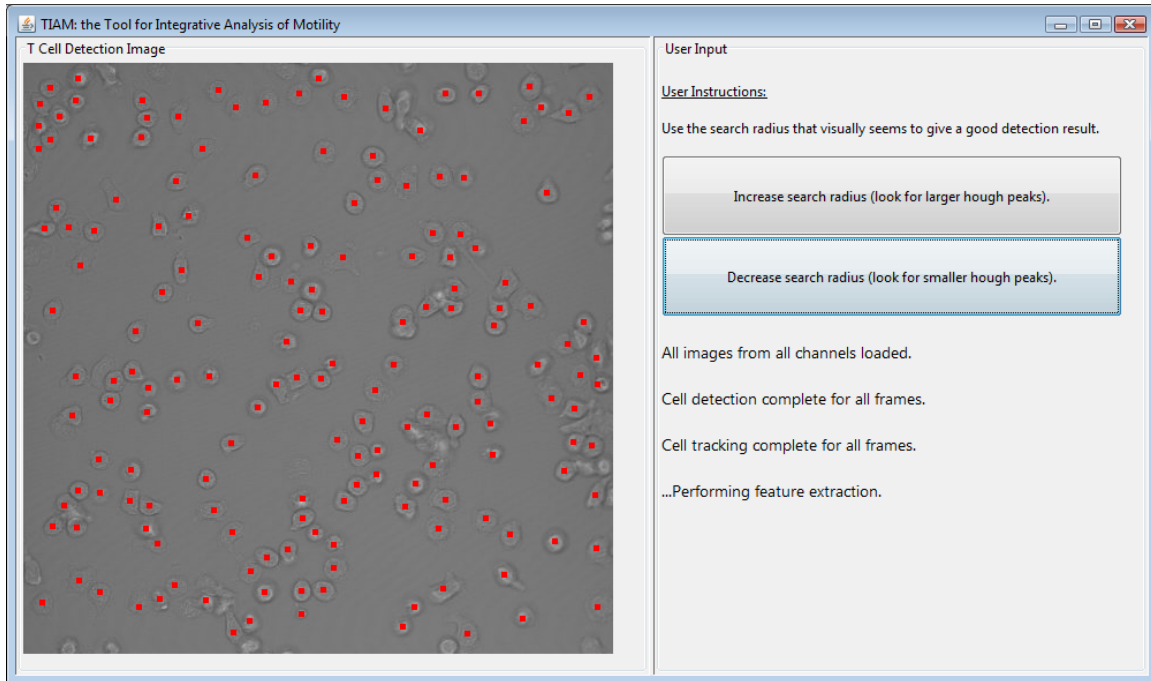
The results of detection for this frame are shown in this final image.

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:



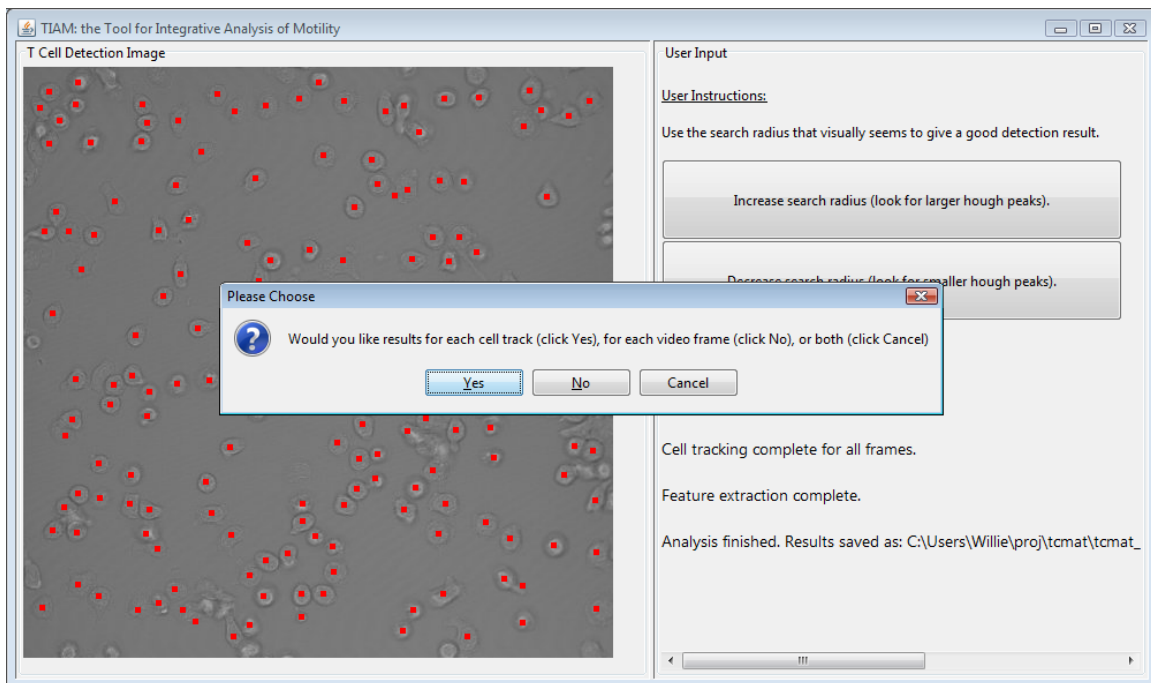
After user confirmation, analysis will be carried out on all of the frames in the video. This analysis can take varying lengths of time, and is mostly dependent on the size of each frame, the number of cells in the experiment, and the number of frames in the video.

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:

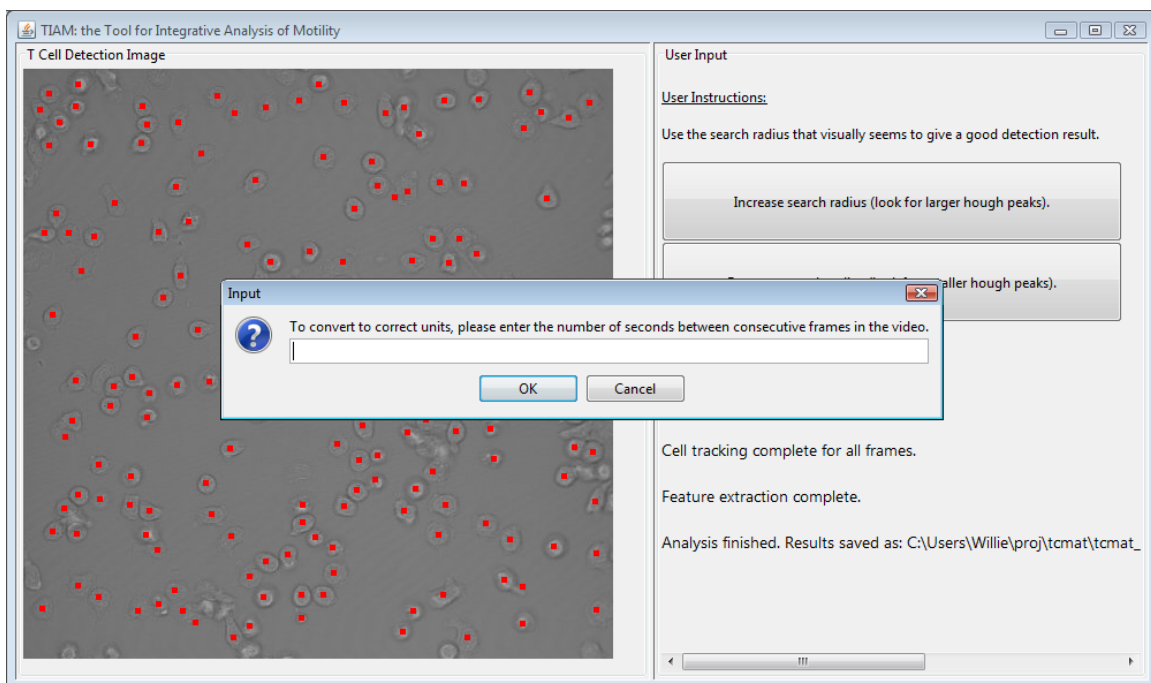


The user is then given the choice of getting motility analysis results for each cell track, for each video frame (averaged over all cells), or both, as shown in the figure below:

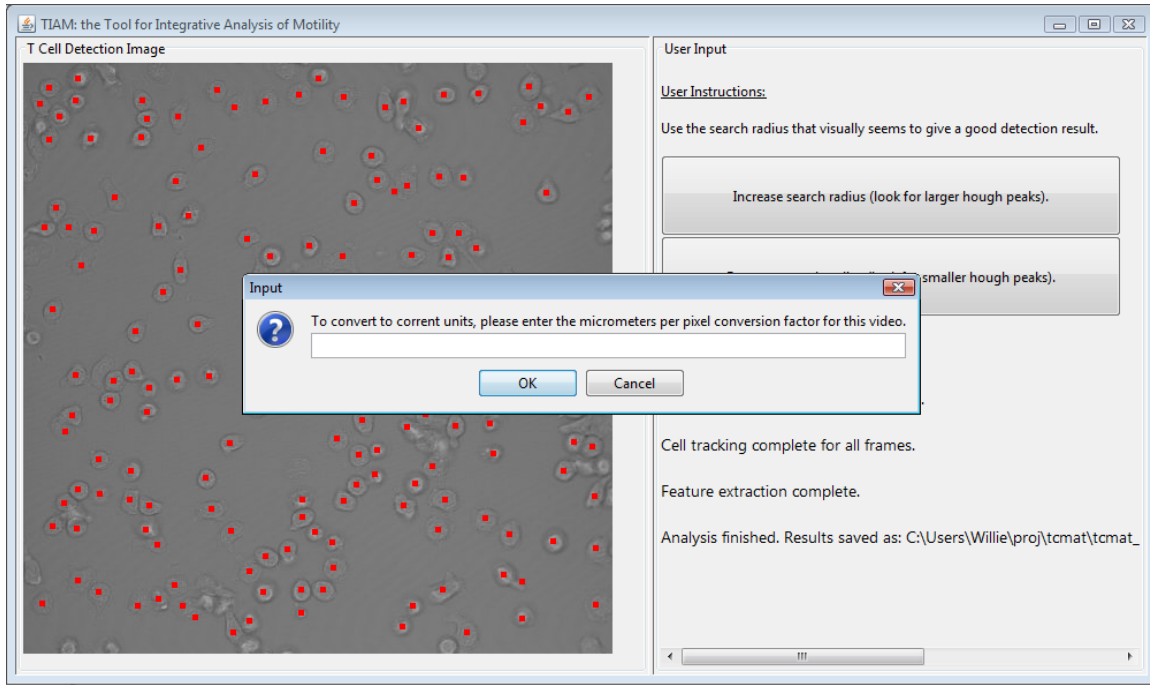




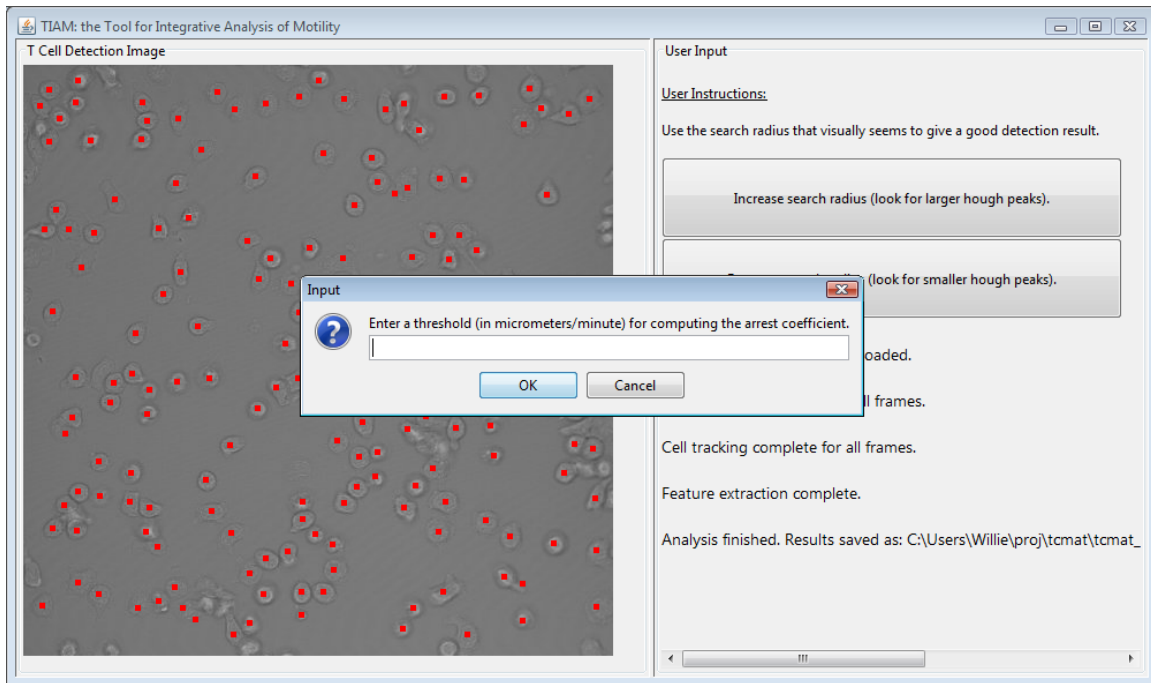
The user is then prompted to enter the number of seconds between frames in the time lapse microscopy video. This allows for speeds and other motility features to be correctly calculated. Entering a value of 20 here suffices for the demoVideo.



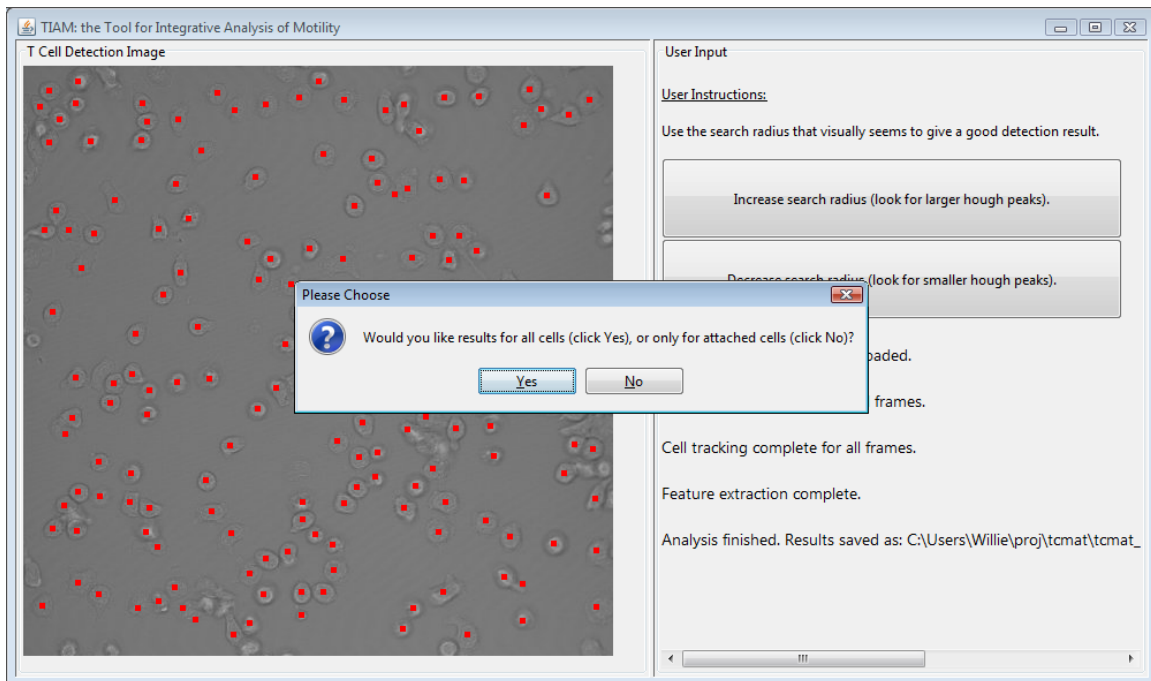
The user is then prompted to enter the size-conversion factor (the number of micrometers per pixel) for each of the time lapse microscopy images. This allows for speeds, cell areas, and other motility features to be correctly calculated. Entering a value of 0.439 here suffices for the demoVideo.



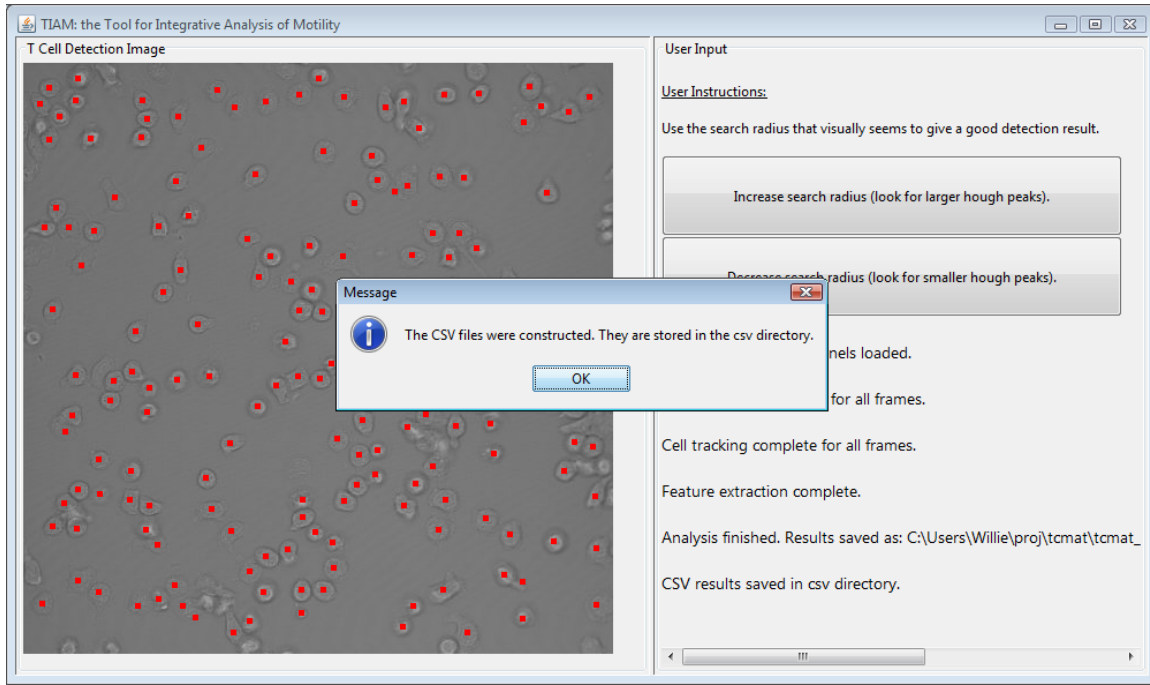
The user is then prompted to enter a threshold for computation of the arrest coefficient (a useful motility feature). Entering a value of 1.3 here suffices for the demoVideo.



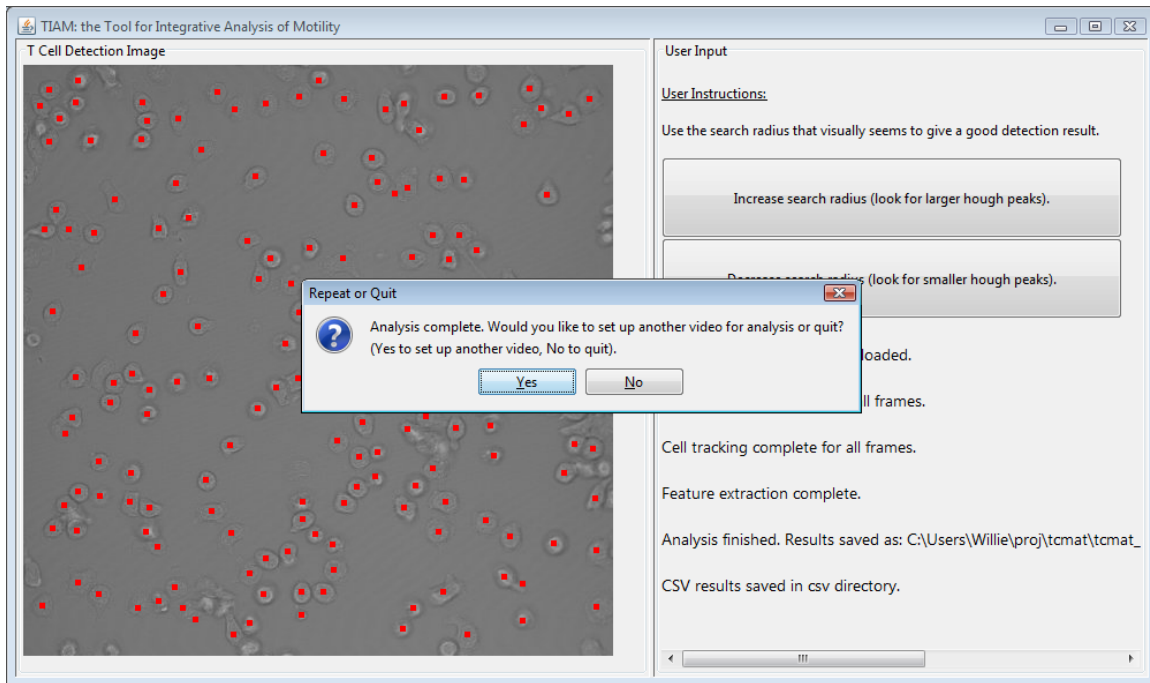
The user can the choose whether to extract data for each cell, or only cells that have attached to some underlying substrate (i.e. cells whose IRM attachment area is non-zero).



After getting all necessary information, TIAM computes the motility results and saves them in CSV format. The CSV files are placed in the TIAM/TIAM\_v01/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/TIAM\_v01/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. 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:



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 \*
- 17-40: outline
- 41: 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

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.

### **Batch script for analyzing multiple experiments**

A batch script has been included in the TIAM repository that allows for analysis of multiple videos without requiring manual setting of detection parameters or interaction with the GUI. The example batch submission script is provided by the function batchSetup.m found in the TIAM\_v01/src/matlab directory. To use this function, a user needs to specify detection parameter values (using this function

as a template), and then run the tcmatBatchMain.m script in the TIAM\_v01/src/ directory. We have found, for a 40x lens with NA 1.3, using a video of attached primary T cells, a reasonable set of detection parameters are:

```
imageScale = 1.2  
edgeValue = 0.1  
radiusMin = 3  
radiusMax = 13  
gradientThresh = 10  
searchRadius = 16  
minCellSeparation = 5  
darkImage = 0
```

### **Important parameters for TIAM developers**

Those wishing to develop and improve upon 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: 367

Variable: max\_trackingjump

Threshold for segment joining

File: joinSubtracks.m

Line: 79

Variable: joinThresh

Bounding box area for IRM attachment extraction

File: tcmatAnalyzeVideo.m

Line: 402

Variable: cropsize\_irm

Histogram normalization for feature extraction

File: tcmatAnalyzeVideo.m

Lines: 411 and 412

Variables: fluor\_adj\_param1 and fluor\_adj\_param2

## 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

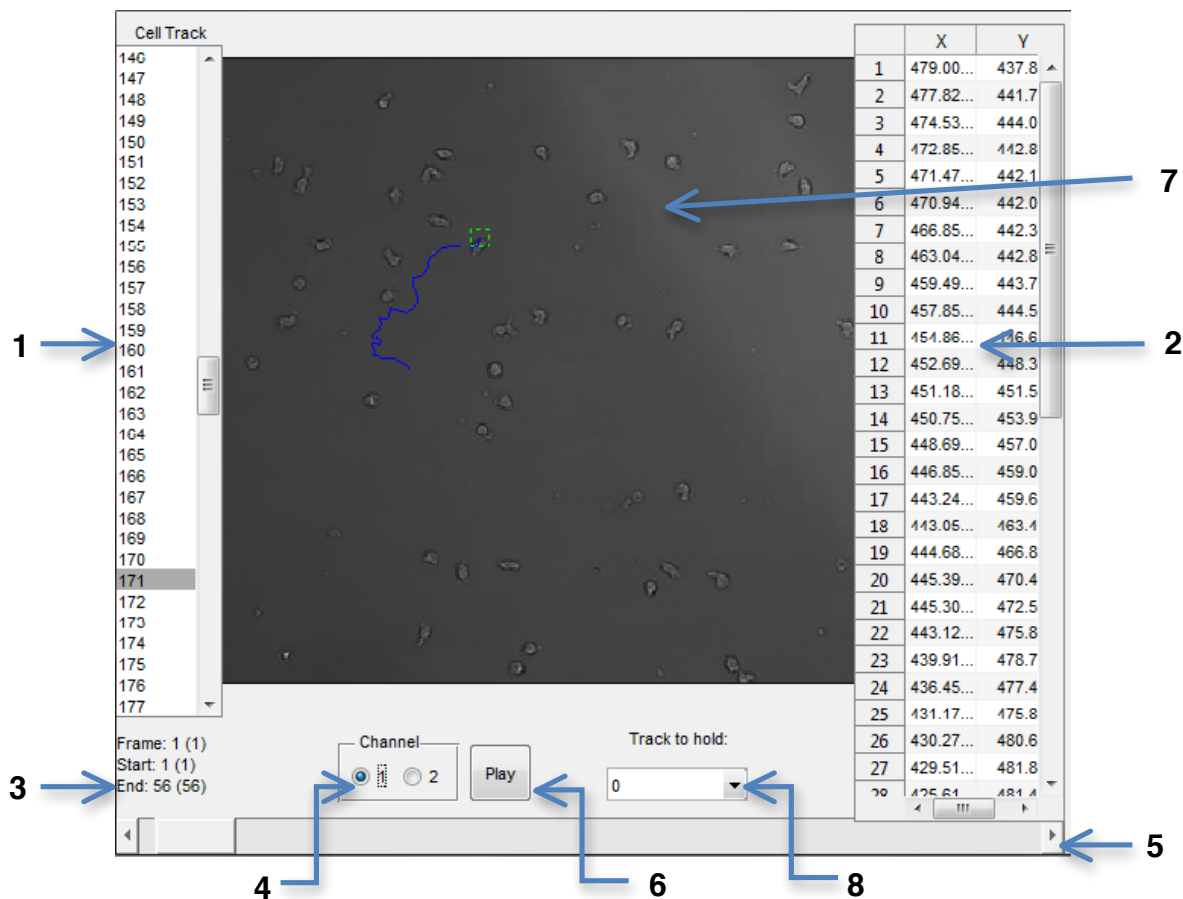
Code for dynamic visualization of tracks is stored in TIAM\_v01/src/trackGUI/

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 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  $\mu\text{m}$  of the horizontal dimension of the images.
- **handles.frameSizeY** – this should be the size in  $\mu\text{m}$  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. Displays 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.

