# eDetect

a fast **E**rror **Dete**ction and **C**orrection **T**ool
for Live Cell Imaging Data Analysis

# User Guide

Hongqing Han

Cell Signaling Dynamics Group

Max Planck Institute for Molecular Genetics, Berlin

January 2019

# Table of Contents

# 1. Introduction

For first-time users, the best way to get to know the usage is to watch the video tutorial (https://www.youtube.com/playlist?list=PLKXmBvkhONktcOXp78igYyeZsCr0wJZg8), which is based on the dataset HaCaT-FUCCI (https://github.com/Zi-Lab/eDetect). So afterwards you can start analyzing the dataset HaCaT-FUCCI following the instructions in Section 6.1. When you have difficulties you can refer to Sections 2-4 for more details.

After these, probably you are already familiar with eDetect live-cell imaging analysis workflow (Section 3.1). So we recommend you to read through Sections 2-4 and Section 6.2 to know more about the functionality.

When you start analyzing your own data, Section 2 and Section 5 may be helpful in the configuration of your own project.

If you have troubles, you can try solving them with the help of Section 7. You are also welcome to let us know the questions you have or the bugs you find. Our email addresses can be found in Section 8.3.

**Font colors of this user guide**

- **Blue**: references to sections, figures, live-cell imaging analysis modules, Graphical User Interfaces (GUIs), GUI components and videos
- **Red**: notes and remarks
- **Orange**: data files

# 2. Getting started

## 2.1. Availability

eDetect source code, standalone executable, and example datasets are available at https://github.com/Zi-Lab/eDetect.

## 2.2. Compatibility and dependencies

- **Option 1**: run eDetect source code in MATLAB (The MathWorks, Inc).

Running eDetect source codes requires MATLAB and Image Processing Toolbox. Parallel computing is possible only if Parallel Computing Toolbox is installed. eDetect source code has been developed and extensively tested with MATLAB 2016b, Image Processing Toolbox (v9.5), Optimization Toolbox (v7.5) and Parallel Computing Toolbox (v6.9) on Windows 7 (64 bit). It has also been tested with MATLAB 2017b, Image Processing Toolbox (v10.1), Optimization Toolbox (v8.0) and Parallel Computing Toolbox (v6.11) on macOS High Sierra 10.13.5 (64 bit).

- **Option 2**: run eDetect standalone executable on Windows.

Running eDetect from standalone executable requires MATLAB Compiler Runtime R2016b (9.1) and Windows 64 bit systems. eDetect standalone executable was compiled on Windows 7 (64 bit), and has been tested on Windows 7 and Windows 10 (64 bit).

## 2.3. Run eDetect

- **Option 1**: run eDetect source code in MATLAB.

Extract the files from the downloaded source code package. Open the folder containing eDetect.m in MATLAB. Run eDetect.m by entering eDetect in command window or by opening the file eDetect.m and clicking Run in the EDITOR tab. The Main Interface appears when eDetect is started (Figure 2.1).

- **Option 2**: run eDetect standalone executable.

Extract the files from the downloaded standalone executable package. Run the executable file. The Main Interface appears when eDetect is started (Figure 2.1).

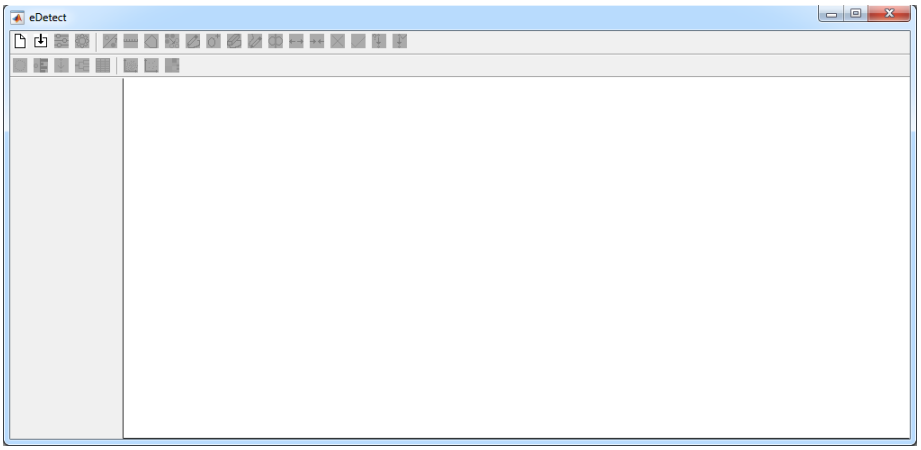



Figure 2.1. Main Interface appears when eDetect is started.

## 2.4. Create a new project

To create a new project:

a. Click New project on Main Interface (Figure 2.1).
b. New Project window (Figure 2.2) appears.
c. Set Directories.
d. Set Filenames.
e. Click button Save to project file.

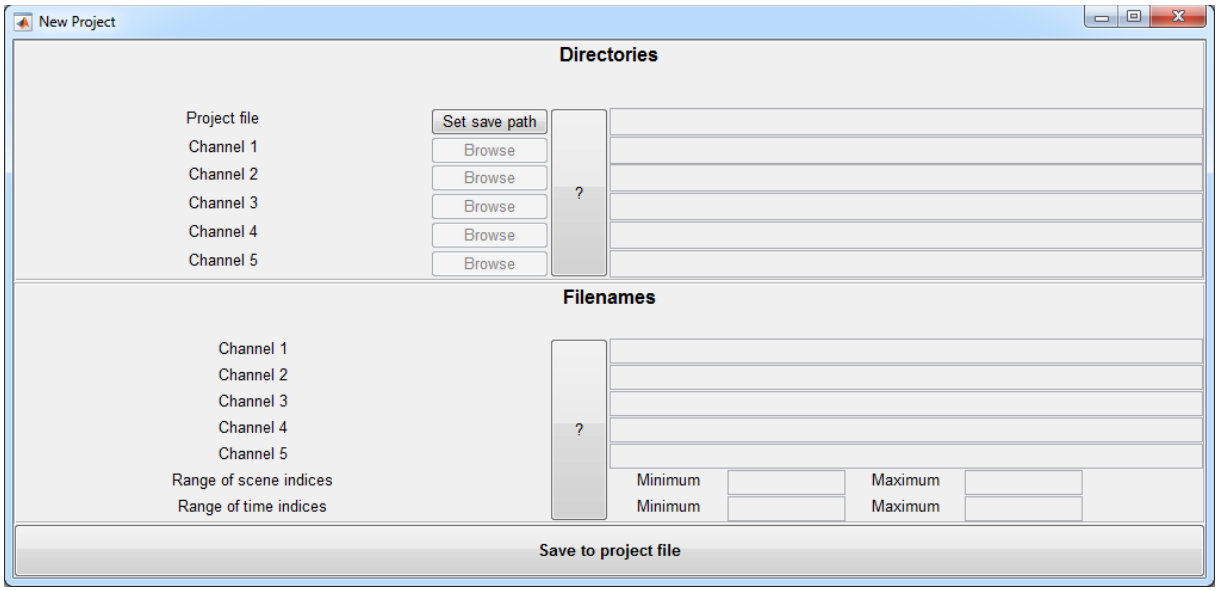



Figure 2.2. Create a new project.

**Directories**

- **Project file:** path of the eDetect project file (.eDetectProject) to be created.
- **Channel 1:** the folder containing Channel 1 images.
  Note: Channel 1 images are used for Cell Segmentation and Feature Extraction (see Section 3.1).
- **Channels 2-5:** the folders containing Channels 2-5 images.
  Note: Channels 2-5 images are used for Measurement (see Section 3.1).

**Filenames**

eDetect is able to analyze grayscale TIFF (.tiff/.tif), JPEG (.jpeg/.jpg), BMP and PNG files.

- **Channel 1:** a filename template for Channel 1 images.
  Note: It contains "<"s and ">"s, which are placeholders of scene and time indices, respectively. The number of "<"s is equal to the number of characters (digits) used by scene indices, and the number of ">"s is equal to the number of characters (digits) used by time indices.
- **Channels 2-5:** filename templates for Channels 2-5 images.
  Note: The format is similar to that of Channel 1.
- **Range of scene indices:** the minimum and maximum of scene indices.
- **Range of time indices:** the minimum and maximum of time indices.

**An example for setting filenames information**

A dataset has only Channel 1 images, which are named:

| |
|---|
| XXXs04t000.tif |
| XXXs04t001.tif |
| … |
| XXXs04t111.tif |

First way to set the filenames information:

a. Channel 1: “XXXs04t>>>.tif”
b. Channel 2: “”
c. Channel 3: “”
d. Channel 4: “”
e. Channel 5: “”
f. Range of scene indices: Minimum “” Maximum “”
g. Range of time indices: Minimum “0” Maximum “111”

An alternative way to set the filenames information:

a. Channel 1: “XXXs<<t>>>.tif”
b. Channel 2: “”
c. Channel 3: “”
d. Channel 4: “”
e. Channel 5: “”
f. Range of scene indices: Minimum “4” Maximum “4”
g. Range of time indices: Minimum “0” Maximum “111”

Note: More examples are provided in Section 6.

## 2.5. Load an existing project

To load an existing project:

a. Click Load project on Main Interface (Figure 2.1).
b. A new window (Figure 2.3) appears. The user can select and open an eDetect project file (.eDetectProject).

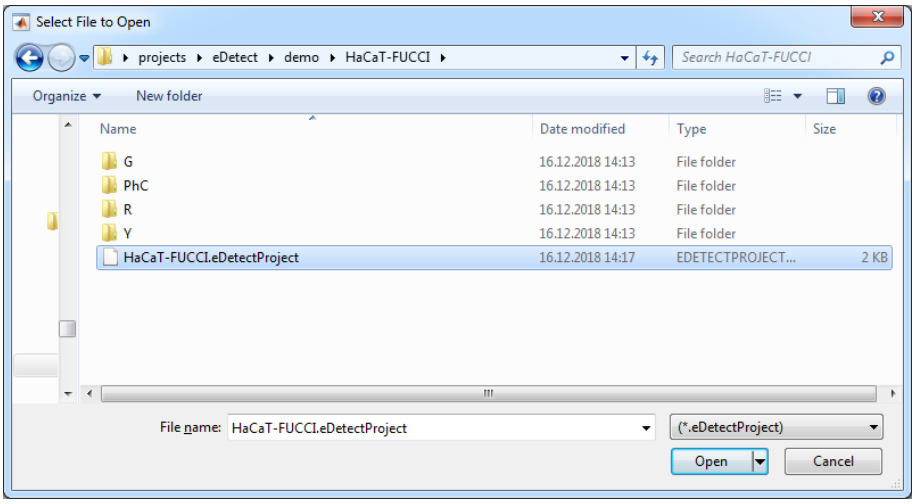



Figure 2.3. Select and open an eDetectProject file.

## 2.6. Specify parameters

To specify parameters:

a. Click Parameters on Main Interface (Figure 2.1).
b. Directories, Filenames and Parameters window appears (Figure 2.4).
c. Fill out Parameters.
d. Click Save to project file.

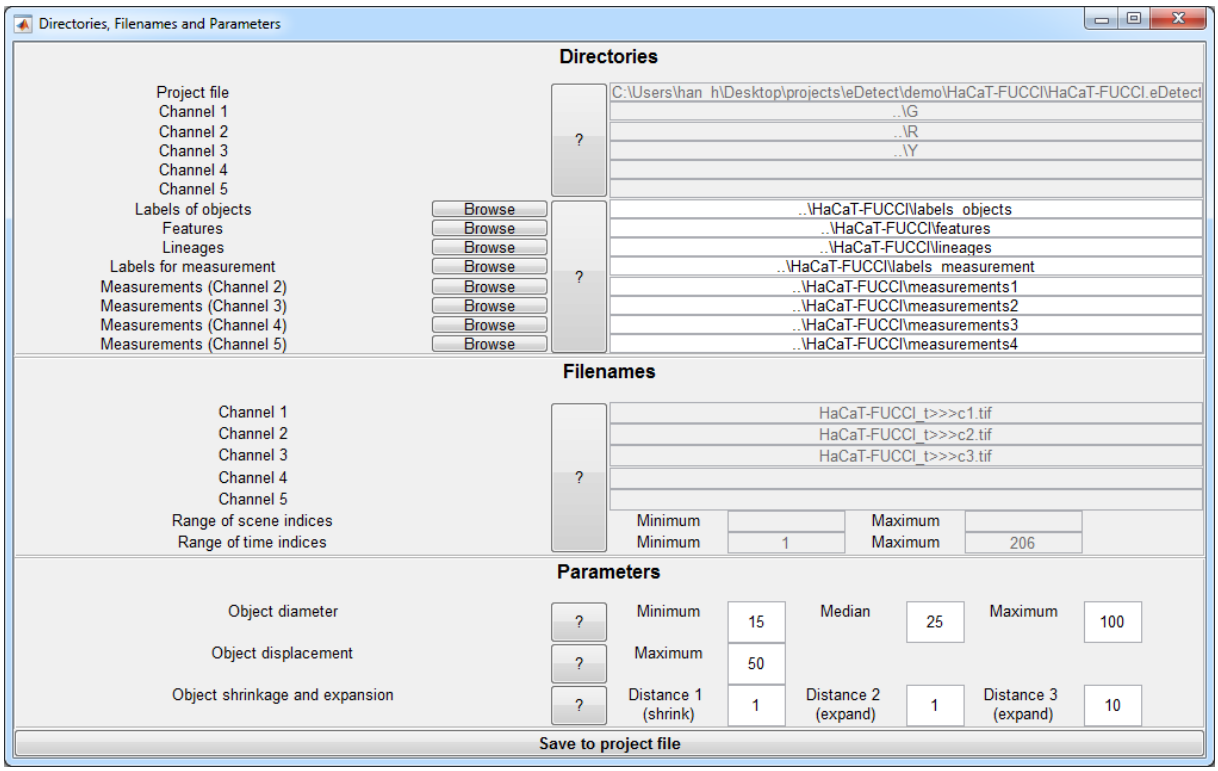



Figure 2.4. Specify parameters.

**Directories**

The directories listed below are used for storing the results of eDetect modules (see Section 3.1). They are located in a folder named under the eDetect project file. They are automatically generated and can be edited manually.

- **Labels of objects:** Results of Cell Segmentation.
- **Labels for measurements:** Results of Measurement.
- **Features:** Results of Feature Extraction.
- **Lineages:** Results of Cell Tracking and Cell Lineage Reconstruction.
- **Measurements:** Results of Measurement.

**Parameters**

All the parameters are lengths. They can be measured using the tool Measure distance in the toolbar on Main Interface. The unit is the pixel.

**a. Object diameter**

- Minimum: diameter of a circle that has an equivalent area to the smallest object.
  Note: If this value is too small some debris will be identified as nuclei, and if it is too large some small nuclei will be discarded.
- Median: diameter of a circle that has an equivalent area to a medium-sized object.
  Note: Usually you can approximate this parameter using the mean of the length and width of the minimum bounding rectangle of a typical nucleus.
- Maximum: diameter of a circle that has an equivalent area to the largest object.
  Note: If this value is too small some large nuclei will be discarded.

**b. Object displacement**

- Maximum: maximum distance an object can move between two consecutive frames.
  Note: Usually you can set a value much larger than all the readouts.

**c. Object shrinkage and expansion**

These parameters are used for measuring Channels 2-5 fluorescence intensities (pixels values). They are required only if Channel 2 images are available.

- Distance 1 ($d_1$): number of pixels by which to shrink the object label (see Section 3.1) in order to generate $c_1$ (Figure 2.5).
  Note: This value is usually small.
- Distance 2 ($d_2$): number of pixels by which to expand the object label (see Section 3.1) in order to generate $c_2$ (Figure 2.5).
  Note: This value is usually small.
- Distance 3 ($d_3$): number of pixels by which to expand the object label (see Section 3.1) in order to generate $c_3$ (Figure 2.5).
  Note: This is the minimum distance between cell membrane and nuclei contour.

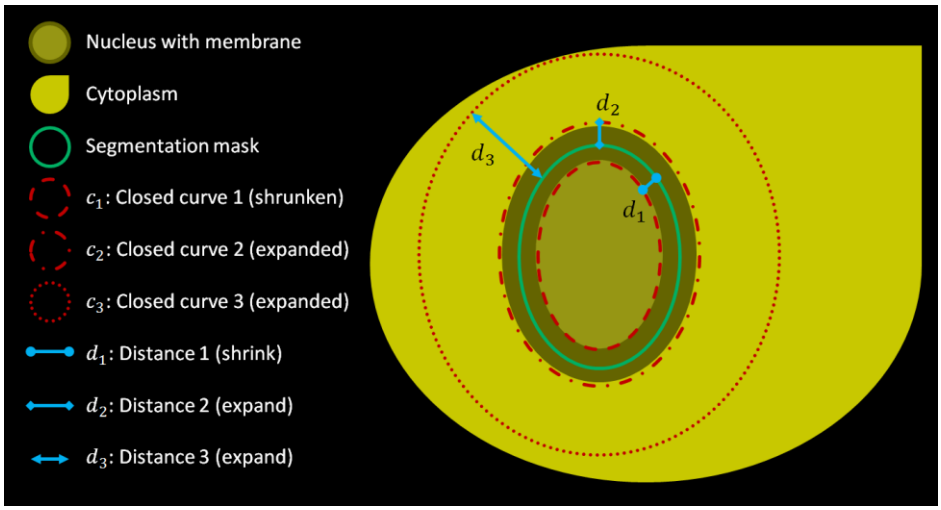



Figure 2.5. An explanation for the parameters: object shrinkage and expansion

Nuclei intensities are measured using the regions within $c_1$. Cytoplasm intensities are measured using the regions between $c_2$ and $c_3$. The purpose of using $c_1$ and $c_2$ is to exclude the nuclear membrane areas, which usually have different brightness and texture from nucleus internal areas.

Note: If you are only interested in measuring Channel 2-5 intensities in nuclei, you can fill out $d_1$ and leave $d_2$ and $d_3$ empty.

# 3. Main Interface

The Main Interface (Figure 3.1) is made up of: image display, navigation panel, display control panel and toolbar.

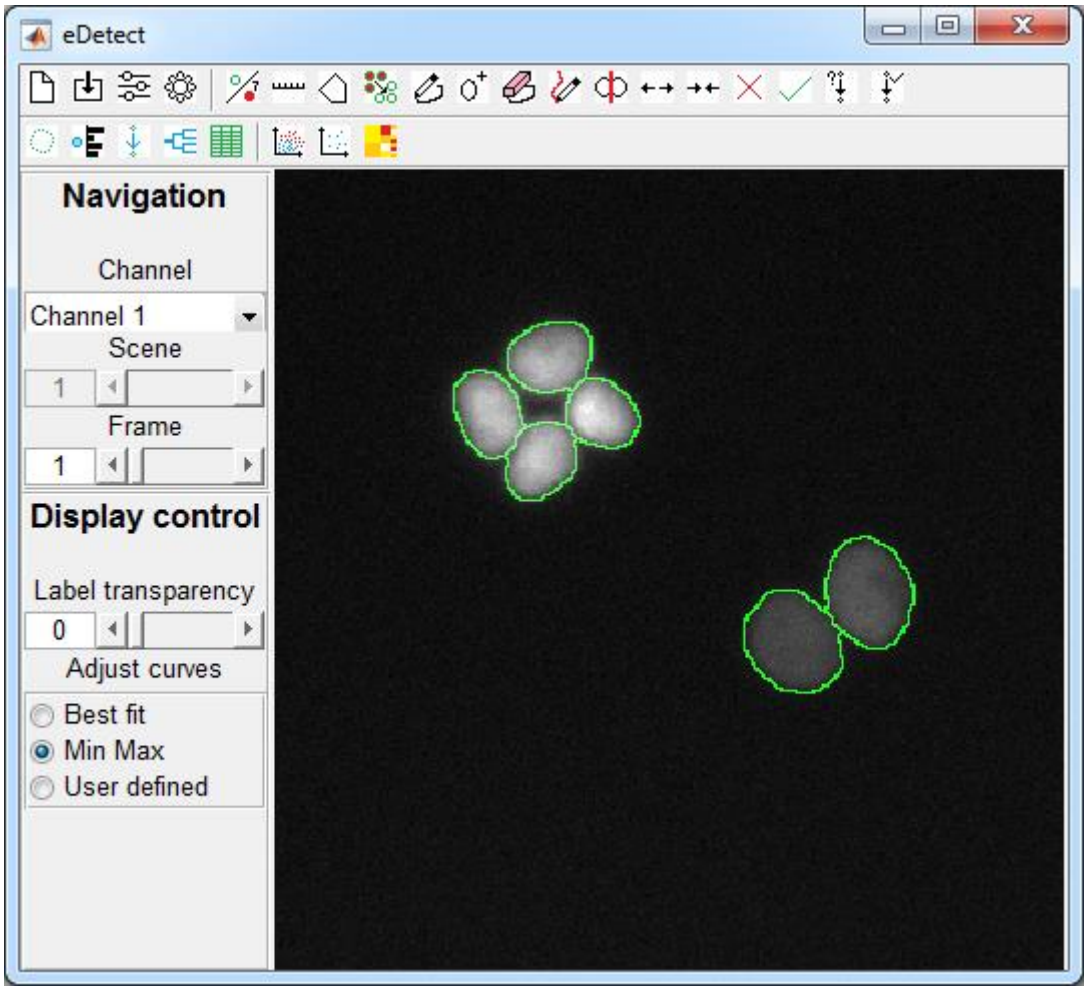



Figure 3.1. Main Interface

## 3.1. Live cell imaging analysis workflow

After creating/loading a project and specifying the parameters, eDetect is ready to start live cell imaging analysis. We recommend the users to run the modules in the following order: Cell Segmentation → Feature Extraction → Segmentation Gating → Cell Tracking → Cell Pair Gating → Cell Lineage Reconstruction → Cell Lineages Display → Measurement. You can find the links to these modules on Main Interface (Figure 3.1).

**Automatic data analysis**

- **Cell Segmentation**
  Input: Channel 1 images.
  Output: object labels.
  Note: In this module, the foreground (bright) objects are identified. Their locations and contours are recorded by object label images. In an object label image, for each foreground object, the value of every pixel within the object contour is equal to the ID of the object. The values of background pixels are 0s.
- **Feature Extraction**
  Input: Channel 1 images and object labels.
  Output: features (an object's coordinate, shape, texture, etc.).
- **Cell Tracking**
  Input: features.
  Output: tracks (an object's predecessor in its preceding frame).
- **Cell Lineage Reconstruction**
  Input: tracks.
  Output: lineages (a matrix whose rows are cells, columns are frames and entries are the objects' IDs in the corresponding frames).
- **Measurement**
  Input: Channels 2-5 images and object labels.
  Output: measurements (mean and median pixel values in nuclei and cytoplasm).

The results of these modules are stored in the folders shown in Section 2.6 (Specify parameters).

**Interactive data visualization**

- **Segmentation Gating**
  Input: features.
  This module will be explained in Section 4.1.
- **Cell Pair Gating**
  Input: features and tracks.
  This module will be explained in Section 4.2.
- **Cell Lineages Display**
  Input: features, measurements and lineages.
  This module will be explained in Section 4.3.

## 3.2. Image display

**Image display**

Image display is used to display raw images with overlaid object labels (see Section 3.1). It is located in the bottom-right of the Main Interface (Figure 3.1).

**Human interactions on image display**

- Left mouse click on object: select one object and highlight it in bright red.
- Right mouse click on object: deselect one object and remove its highlight.

**Navigation panel**

- **Channel:** a navigator for selecting the channel of the images.
- **Scene:** a navigator for selecting the scene of the images.
- **Frame:** a navigator for selecting the frame of the image.

**Display control panel**

- **Label transparency:** transparency of object labels that are overlaid onto raw image.
- **Adjust curves:** image tonality adjustments for improving brightness or contrast.
  Pixel values are adjusted with linear transformation $[l_0, h_0] \rightarrow [l_1, h_1]$. For example, for 8-bit images $l_1 = 0$ and $h_1 = 255$, while for 16-bit images $l_1 = 0$ and $h_1 = 65535$. Pixels values out of range $[l_0, h_0]$ will be saturated to $l_1$ or $h_1$. Suppose the image is $I$.
  - User defined: $l_0$ and $h_0$ need to be specified by user.
  - Min Max: $l_0 = \min(I)$ and $h_0 = \max(I)$. No pixels are saturated.
  - Best fit: $l_0 = P_1(I)$ and $h_0 = P_{99}(I)$ in which $P_i$ is $i$th percentile. "Best fit" is similar to "Min Max" except that the brightest 1% and the dimmest 1% pixels are saturated.

**Image display tools**

**Change overlay**
This is used for the display of object labels. Switch between displaying contours (green/red) and labels (colored masks with index numbers).

**Measure distance**
This is used to specify parameters (Section 2.6). The parameters are measured in pixels.

**Video tutorial**

We made a video to show how to configure image display. It is "6 Main Interface" in the list "eDetect tutorial":
https://www.youtube.com/watch?v=X1qYUT65SL0&t=0s&list=PLKXmBvkhONktcOXp78igYyeZsCr0wJZg8&index=8.

## 3.3. Manual correction of segmentation and tracking errors

eDetect Main Interface (Figure 3.1) supports manual corrections of Cell Segmentation and Cell Tracking results.

**Manual correction tools**

- **Draw a polygon**
  Objects whose centroids are in the polygon will be selected (highlighted).
- **Deselect all**
  Deselect all selected objects (remove highlight).
- **Draw an object (keyboard "1")**
  Use mouse to draw a closed curve.
- **Create objects (keyboard "c")**
  The closed curves will become the contours of new objects.
- **Remove objects (keyboard "r")**
  The area occupied by the selected objects will become background.
- **Draw division (keyboard "2")**
  Draw an open curve to be used as a division line.
- **Divide objects (keyboard "d")**
  The open curve will become background. Therefore some objects will be divided into parts.
- **Split objects (keyboard "s")**
  Split the selected objects into parts.
- **Merge objects (keyboard "m")**
  Merge the selected objects into one object.
- **Delete objects (keyboard "l")**
  Label selected objects as erroneous objects (their green contours become red). They won't be considered in Cell Tracking or Cell Lineage Reconstruction. However, they are still considered objects and will be used to generate foreground region.
- **Recover objects (keyboard "v")**
  Recover selected objects from erroneous state and make them normal objects again. They will be considered in Cell Tracking and Cell Lineage Reconstruction.
- **Get predecessor**
  After selecting one object and clicking Get predecessor, the previous frame will be displayed and the object's predecessor will be highlighted.
- **Set predecessor**
  After Get predecessor, if the user deselect the highlighted predecessor, select another object, and click Set predecessor, the new selected object will become the new predecessor. If no object is selected before clicking Set predecessor, the object selected before clicking Get predecessor will have no predecessor.

**Video tutorial**

Video "6 Main Interface" in the list "eDetect tutorial":
https://www.youtube.com/watch?v=X1qYUT65SL0&t=0s&list=PLKXmBvkhONktcOXp78igYyeZsCr0wJZg8&index=8.

# 4. Error detection and correction

## 4.1. Segmentation Gating

To start Segmentation Gating module (Figure 4.1), click  on Main Interface (Figure 3.1). Segmentation Gating uses Principal Component Analysis (PCA) to visualize single cell nuclei morphological features, and helps users to detect and correct segmentation errors.

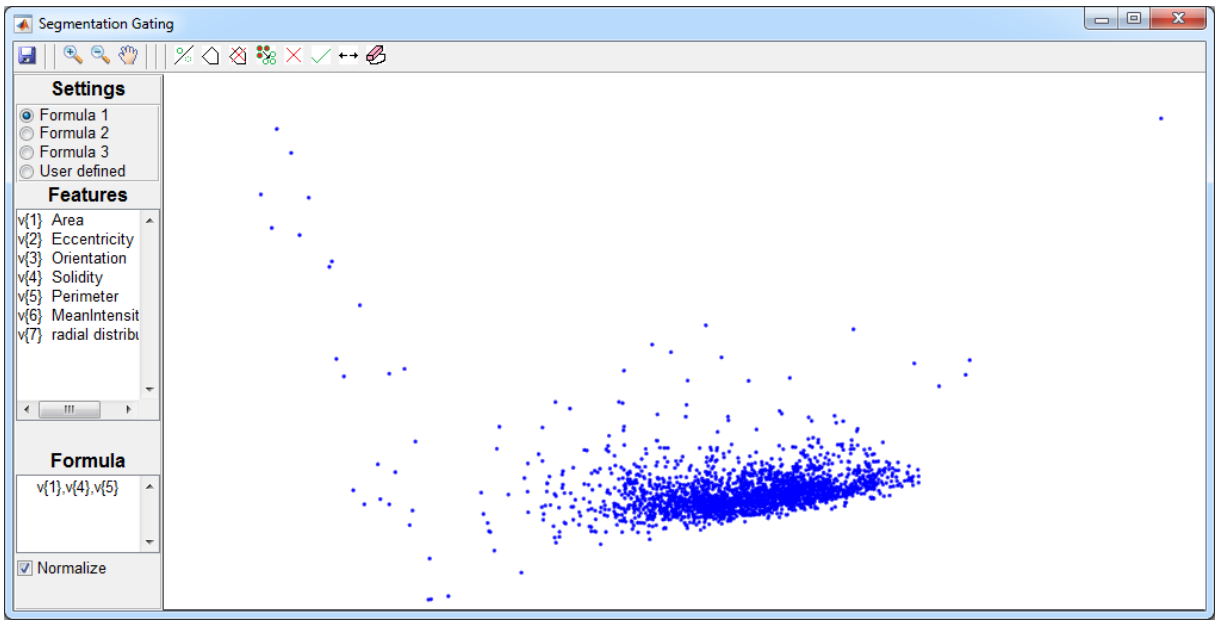



Figure 4.1. The Segmentation Gating window.

**Scatter plot**

Each point represents an object. The axes of the scatter plot are the 1st and 2nd principal components of the PCA.

**Features**

The meanings of features from $v\{1\}$ to $v\{6\}$ are explained at MathWorks website (https://www.mathworks.com/help/images/ref/regionprops.html). The 7th feature "radial distribution" describes how intensity varies as a function of distance from the nearest point in the border of the object. For more details please refer to the source code.

**Formula**

Operators

- Addition: "+"
- Subtraction: "-"
- Multiplication: ".*"
- Division: "./"
- Exponentiation: ".^"

**Settings**

a. Formula 1: 'v{1},v{4},v{5}'.
   This is the default setting for Segmentation Gating. For many cell types, the shapes of nuclei are close to a circle or an ellipse, and their borders are relatively smooth. While segmentation errors (e.g. under-segmented nuclei clumps, over-segmented nuclei area or bright background area) may have different shapes or have less smooth contours. Therefore the default setting is helpful in finding segmentation errors for different types of cells.
b. Formula 2: 'v{1},v{2}'.
   This is the default setting for Cell Pair Gating. The features “Area” and “Eccentricity” are useful in separating newly divided sister cells, which are often smaller and slimmer than nuclei in other phases.
c. Formula 3: 'v{6},v{7}'.
   This could also be useful in separating newly divided sister cells, which are often brighter (especially in the center of the objects) than other nuclei.
d. User defined
   eDetect allows the users to customize PCA input variables with the following steps:
   a) Select User defined.
   b) Select (click) features in Features that you want to include.
   c) Click Generate formula (will appear).
   d) Formula is updated according to the features you selected.
   e) Customize Formula using operators “+”, “−”, “.*”, “./” or “. ^” (optional).
   f) Select or deselect Normalize (optional).
   g) Click Update (will appear).
   h) Scatter plot is updated.

**Toolbar: MATLAB tools**

- Save figure
- Zoom in: left-click or draw a rectangle on scatter plot.
- Zoom out: left-click on scatter plot.
- Move plot: drag scatter plot with mouse left button.

**Toolbar: eDetect tools**

The eDetect tools here are the same with their counterparts in Main Interface, except:

- **Display deleted**
  This is a toggle tool. Click to also display deleted objects.
- **Draw a polygon**
  Draw a polygon on scatter plot. The points inside the polygon will be selected (highlighted red).
- **Delete polygons**
  Remove all the polygons from scatter plot.
- **Clear selection**
  Deselect all selected points and remove all polygons.

**Human interactions**

Mouse clicks on scatter plot in Segmentation Gating.

- **Left mouse click**
  Select and highlight the closest point. The object it represents will be selected and highlighted on image display in Main Interface.
- **Right mouse click**
  Deselect the closest point and remove its highlight. The object it represents will be selected and highlighted on image display in Main Interface.

Mouse clicks on image display in Main Interface.

- The points representing objects selected on Main Interface will be marked by a red square in the scatter plot in Segmentation Gating.

**Usage**

Objects with similar appearances are often located close to each other. Therefore different areas in the scatter plot may represent different sub-populations.

There are two strategies to find out about the locations of different sub-populations:

- Click on the scatter plot to inspect the object on image display.
- Select objects on image display to see its location in the scatter plot.

**Video tutorial**

We made a video to show how to customize the scatter plot. It is "7 Scatter plots" in the list "eDetect tutorial":
https://www.youtube.com/watch?v=DQS96k4qHHg&index=8&list=PLKXmBvkhONktcOXp78igYyeZsCr0wJZg8.

## 4.2. Cell Pair Gating

To start the Cell Pair Gating module (Figure 4.2), click on Main Interface (Figure 3.1). Cell Pair Gating uses PCA to visualize morphological features of object pairs (two objects having the same predecessor), and helps users to detect and correct segmentation and tracking errors.

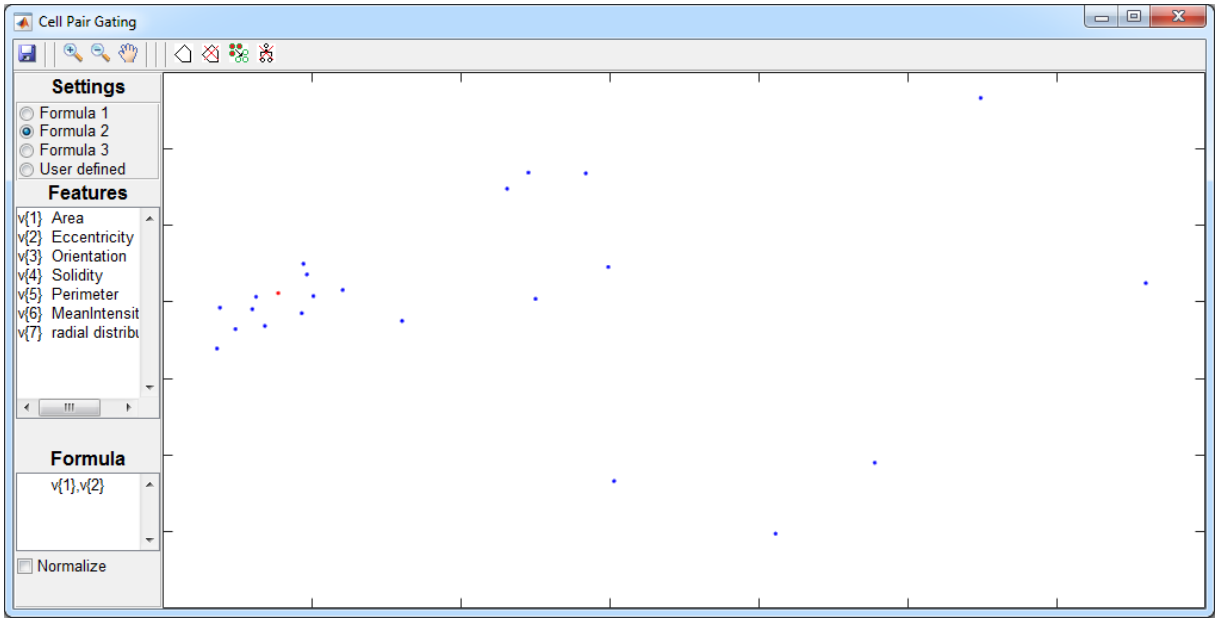



Figure 4.2. Cell Pair Gating window.

**Scatter plot**

Each point represents a pair of objects that have the same predecessor in the preceding frame. Ideally this means the two nuclei are from two daughter cells with the same mother cell. However, this can also result from segmentation or tracking errors. Here, all the "divisions" are visualized so that users can click on a point to see whether it represents a true cell division event or not. If there is error, the user can correct it on Cell Pair Gating or Main Interface.

The axes of the scatter plot are the 1st and 2nd principal components of a PCA dimension reduction. The PCA input variables in Formula can be customized in the same way as in Segmentation Gating.

**Settings**

The default setting is Formula 2 ("$v\{1\}, v\{2\}$"), which means features "Area" and "Eccentricity".

**Toolbar: MATLAB tools**

The MATLAB tools here are the same with their counterparts in Segmentation Gating.

**Toolbar: eDetect tools**

The eDetect tools here are the same with their counterparts in Segmentation Gating, except:

- **Detach daughter cells from mother cell**
  Remove the association between the pair of "daughter cells" and the "mother cell". After clicking this tool, the two cells will have no predecessor in the preceding frame.

**Human interactions**

Mouse clicks on scatter plot on Cell Pair Gating.

- **Left mouse click**
  Select and highlight the closest point. The pair of objects will be selected and highlighted on image display in Main Interface.
- **Right mouse click**
  Deselect the closest point and remove its highlight. The pair of objects will be selected and highlighted on image display in Main Interface.

**Usage**

As similar object pairs are likely located close to each other, different areas in the scatter plot may represent different sub-populations. The user can click on the scatter plot to inspect the pair of objects in Image Display. After clicking a few points, you may figure out where errors are likely located.

## 4.3. Cell Lineages Display

To start the Cell Lineages Display (Figure 4.3), click on Main Interface (Figure 3.1). The interface is made up of: lineage tree heatmap, display control, lineage filters, outlier detection and toolbar.

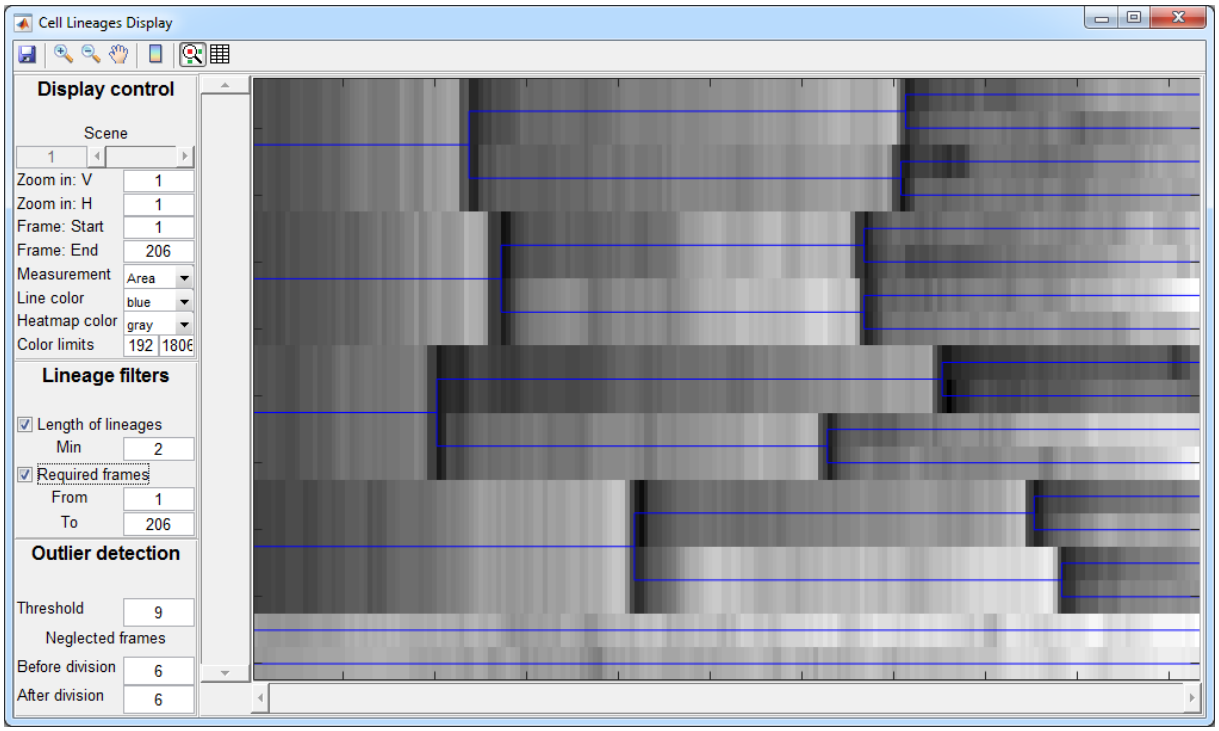



Figure 4.3. Cell Lineage Display window.

**Lineage tree heatmap**

This is used to visualize quantified features (results of Feature extraction, Section 3.1) and measurements (results of Measurement, Section 3.1) in lineages.

- **Data:** The heatmap presents a matrix of user-selected feature or measurement values. The default is object area. The larger the value is, the brighter the corresponding entry of the heatmap is. Each row represents a cell lineage and each column represents a frame.
- **Highlights:** Cell lineages are highlighted by lines. A cell division is indicated by a bifurcation unless one of the daughter cells (one line) is filtered out from display, in which case a vertical line segment is used to mark the division event. Automatically detected outliers are highlighted in red horizontal line segments.

**Display control**

- **Scene:** This is a navigator for selecting the scene.
- **Zoom in: V (Vertical):** Magnify the heatmap vertically. The slider to the left of the heatmap will also be enabled when the number in the text edit is larger than 1.
- **Zoom in: H (Horizontal):** Magnify the heatmap horizontally. The slider below the heatmap will also be enabled when the number in the text edit is larger than 1.
- **Frame: Start:** This is the first frame to display in heatmap. Earlier frames will not be displayed.
- **Frame: End:** This is the last frame to display in heatmap. Later frames will not be displayed.

- **Measurement:** All the features and measurements are listed here. The user can select one of them to visualize using heatmap.
- **Line color:** color of lines that highlight the heatmap**.**
- **Heatmap color:** color map of the heatmap.
- **Color limits:** minimum and maximum of the color map of the heatmap.

**Lineage filters**

- **Length**
  Only the lineages that cover at least $n$ consecutive frames are kept ($n$ is the Min - the minimal length of cell lineage). Other lineages are not displayed.
- **Required frames**
  Only lineages that cover all frames from $f_1$ (From) to $f_2$ (To) are kept (Figure 4.4). Other lineages are not displayed.

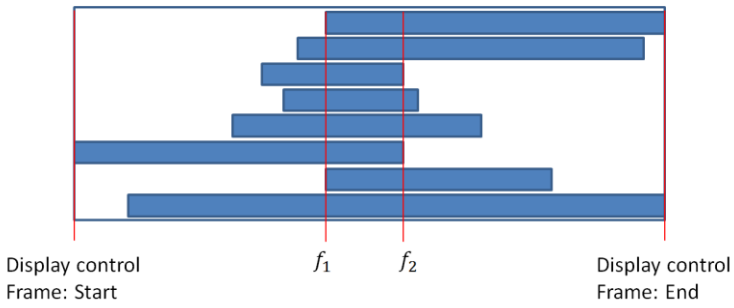



Figure 4.4. An explanation of Required frames.

**Human interactions**

- **Left mouse click on heatmap**
  The object corresponding to the clicked heatmap entry will be selected and highlighted in Image Display on Main Interface (Figure 3.1).
- **Right mouse click on heatmap**
  Open Synchrogram (Figure 4.4) to visualize the cropped images of this cell lineage.

**Toolbar: MATLAB tools**

MATLAB tools here are the same with their counterparts in Segmentation Gating and Cell Pair Gating, except:

- **Color bar**
  A mapping between colors and feature values.

**Toolbar: eDetect tools**

- **Outlier detection**
  Start outlier detection and highlight the outliers in red on heat map.
- **Export table**
  Save the data tables of cell lineages (row: cell lineages, column: frames) with features or measurements (Section 3.1). In the 1st table the entries are the IDs of the cell in the frame, and in the 2nd table the entries are features or measurements.

**Outlier detection**

- **Threshold**
  The number of standard deviations $n$. Points whose distances to the median are larger than $n$ times standard deviation are considered outliers.
- **Neglected frames**
  The number of frames before and after cell divisions to be excluded from outlier detection. When cells divide, they often demonstrate large morphological changes under fluorescent microscopes. Therefore, we exclude this period from outlier detection.

**Usage**

In heatmap, users should pay attention to the following listed patterns, which often indicate errors.

- **Frequent cell divisions**
  This means a cell divided twice within a short time interval. The users should check it based on the normal doubling time of the cells.
- **Outliers**
  Outlier detection marks sudden changes in the visualized feature or measurement values. Many sudden changes are caused by segmentation or tracking errors.
- **Late started lineages (temporarily undetected cells)**
  Suppose the dataset has $n$ frames (from Frame $1$ to Frame $n$), and the lineage exists from Frame $a$ to Frame $b$.
  If $a > 1$, then there could be 3 possible reasons why the lineage doesn't extend to Frame $a - 1$:
  - In Frame $a - 1$, the object is not in the field of view.
  - In Frame $a - 1$, the object is in the field of view but regarded as background.
  - In Frame $a - 1$, the object is identified as foreground, but affected by errors, and therefore excluded from Cell Tracking and Cell Lineage Reconstruction. If errors are corrected, the lineage might extend to Frame $a - 1$ or even earlier.
- **Early ended lineages (temporarily undetected cells)**
  Suppose the dataset has $n$ frames (from Frame $1$ to Frame $n$), and the lineage exists from Frame $a$ to Frame $b$.
  If $b < n$, then there could be 3 possible reasons why the lineage doesn't extend to Frame $b + 1$:
  - In Frame $b + 1$, the object is not in the field of view.
  - In Frame $b + 1$, the object is in the field of view but regarded as background.
  - In Frame $b + 1$, the object is identified as foreground, but affected by errors, and therefore excluded from Cell Tracking and Cell Lineage Reconstruction. If errors are corrected, the lineage might extend to Frame $b + 1$ or even later frames.

## 4.4. Synchrogram

The Synchrogram window (Figure 4.5) appears when the heatmap on Cell Lineages Display (Figure 4.3) is right-clicked. The Synchrogram of a lineage is a montage of cropped images of the cell lineage in each frame. In each cropped image, the selected cell is located in the center. The frame in which the cell is highlighted corresponds to the frame that is right-clicked by the user on heatmap.

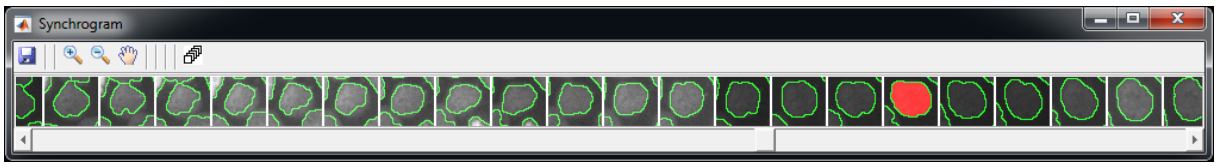



Figure 4.5. Synchrogram windows.

**Toolbar: MATLAB tools**

The MATLAB tools here are the same with their counterparts in Segmentation Gating, Cell Pair Gating, and Cell Lineages Display.

**Toolbar: eDetect tools**

**Export video**
Save the cropped images into the folder where the lineages are stored (Section 2.6).

**Human interactions**

**Left or right mouse click on a cropped image**
The frame will be displayed on image display in Main Interface. The object will be selected and highlighted.

# 5. Advanced settings

To change settings:

a. Click Settings on Main Interface (Figure 3.1)
b. The Settings window (Figure 5.1) appears
c. Change settings
d. Click "Save to project file"

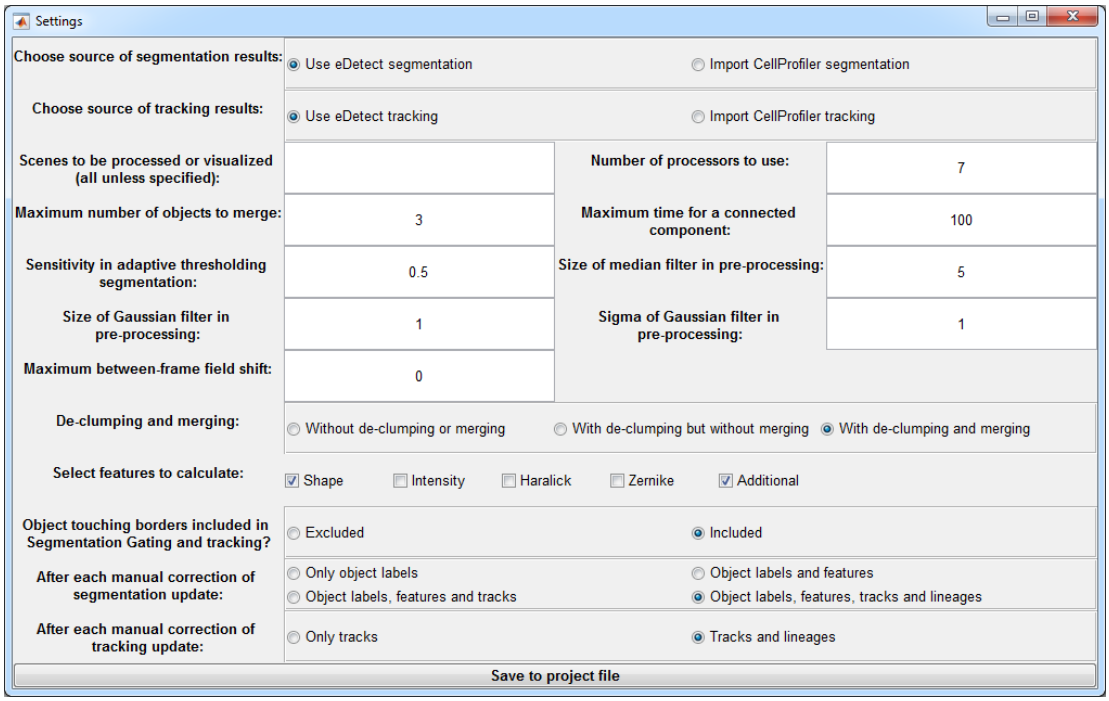



Figure 5.1. Settings window.

- **Choose source of segmentation results:** eDetect provides the option of importing cell segmentation results from CellProfiler. Details are explained in Section 5.1.
- **Choose source of tracking results:** eDetect provides the option of importing cell tracking results from CellProfiler. Details are explained in Section 5.1.
- **Scenes to be processed or visualized (all unless specified):** If you only want a subset of scenes to be processed by an automatic data analysis or interactive data visualization module (Section 3.1), this parameter needs to be specified. It requires an array of the scene indices separated by spaces, e.g. "2 3 4 5 6". If the text edit is left blank, all the scenes are processed by the module.
- **Number of processors to use:** Number of processors used. The default value is set by detecting the number of logical cores in the computer. If there are more than two, e.g. if there are $x$ logical cores, then eDetect sets $x - 1$ as the default value; otherwise the default value is 1. Note that MATLAB Parallel Computing Toolbox is needed for using multiple cores.
- **Maximum number of objects to merge:** In the merging step after de-clumping, eDetect evaluates all the combinations of several connected objects and decide whether to merge them together or not. This parameter is the maximum number of connected objects to be considered for merging at a time. The default value is 3. Larger values can result in fewer over-segmentation errors, but require more time.
- **Maximum time for a connected component:** Maximum number of seconds spent for each connected component (of objects) in the merging step. The unit is the second.

- **Sensitivity in adaptive thresholding segmentation:** MATLAB documentation: "A scalar in the range [0, 1] that indicates sensitivity towards thresholding more pixels as foreground. A high sensitivity value leads to thresholding more pixels as foreground, at the risk of including some background pixels." The default value is 0.5. Please find more about MATLAB adaptive thresholding at "https://www.mathworks.com/help/images/ref/adaptthresh.html".
- **Size of median filter in pre-processing:** Please find more about MATLAB 2-D median filtering at "https://www.mathworks.com/help/images/ref/medfilt2.html".
- **Size of Gaussian filter in pre-processing:** Please find more about MATLAB 2-D Gaussian filtering at "https://www.mathworks.com/help/images/ref/imgaussfilt.html".
- **Sigma of Gaussian filter in pre-processing:** Please find more about MATLAB 2-D Gaussian filtering at "https://www.mathworks.com/help/images/ref/imgaussfilt.html".
- **Maximum between-frame field shift:** When the imaging platform is unstable, there could be spatial shift of field between 2 consecutive frames. Here users need to specify the maximum possible distance of this shift.
- **De-clumping and merging:** The user needs to decide whether to conduct declumping and merging steps after adaptive thresholding in Cell Segmentation.
  - **Without de-clumping or merging:** This works well if objects do not touch each other.
  - **With de-clumping but without merging:** This is faster than the 3rd option but could produce more over-segmentations.
  - **With de-clumping and merging (default):** Both de-clumping and merging will be conducted.
- **Select features to calculate:** Tick features to calculate. Features can be visualized in Segmentation Gating, Cell Pair Gating and Cell Lineages Display only if they are calculated. First two are mandatory. In the current version, Additional features only include "radial distribution".
- **After each manual correction of segmentation update:** When Cell Segmentation results are manually corrected, the downstream results may not be consistent with it anymore. Therefore, the downstream results need to be updated. Here users can decide which results to update. The definitions of object labels, features, tracks and lineages can be found in Section 3.1.
  - **Only object labels:** Only Cell Segmentation results are updated. Other results may require re-running the modules to be compatible with segmentation results.
  - **Object labels and features:** Cell Segmentation results are updated. Results of Features Extraction are updated if already available. Other results may require re-running the modules to be compatible with segmentation results.
  - **Object labels, features and tracks:** Cell Segmentation results are updated. Results of Features Extraction and Cell Tracking are updated if already available. Other results may require re-running the modules to be compatible with segmentation results.
  - **Object labels, features, tracks and lineages (default):** Cell Segmentation results are updated. Results of Features Extraction, Cell Tracking and Cell Lineage Reconstruction are updated if already available. Other results may require re-running the modules to be compatible with segmentation results.
- **After each manual correction of tracking update:** Users can choose whether to update cell lineages or not. The definitions of tracks and lineages can be found in Section 3.1.
  - **Only tracks:** Only Cell Tracking results are updated. Lineages may require re-running Cell Lineage Reconstruction to be compatible with tracking results.
  - **Tracks and lineages (default):** Cell Tracking results are updated. Lineages are also updated if already available.

## 5.1. Importing CellProfiler segmentation and tracking results

**Import segmentation**

To import CellProfiler segmentation results, click Cell Segmentation on Main Interface and select all the segmentation mask files, which have the same names (extensions could be different) as input images.

To generate segmentation masks that have the same filenames as input files, you need to specify the settings in "SaveImages" (CellProfiler module) as in Figure 5.2. The image name "DNA" depends on how you name the input images.

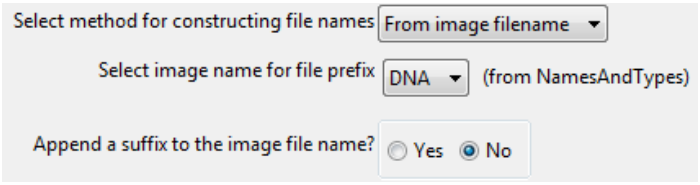



Figure 5.2 CellProfiler module "SaveImages"

**Import tracking**

To import CellProfiler tracking results, tracking results have to be stored in a 3-column csv file with a header. For example (Table 5.1), in a dataset there are 2 scenes (s) and 2 time points (t). In image 1 (s1t1, green) there are 2 cells without predecessors, but they are the predecessors of cell1 and cell2 of image 2 (s1t2, red). In image 3 (s2t1, cyan) there is 1 cell without a predecessor, but it is the parent of the sister cells in image 4 (s2t2, pink).

| ImageNumber,ObjectNumber,TrackObjects_ParentObjectNumber_50 |
|---|
| 1,1,0 |
| 1,2,0 |
| 2,1,1 |
| 2,2,2 |
| 3,1,0 |
| 4,1,1 |
| 4,2,1 |

Table 5.1. An example of CellProfiler tracking result that can be imported by eDetect.

To generate a data table as describe in Table 5.1, you need to "Select the measurement to export" and "Press button to select measurements" in "ExportToSpreadsheet" (CellProfiler module, Figure 5.3).

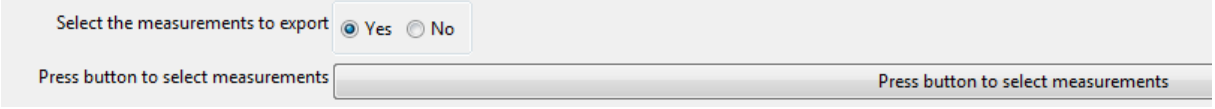



Figure 5.3. CellProfiler module "ExportToSpreadsheet".

After pressing the button, in the window "Select measurements" select the measurement "All - IdentifyPrimaryObjects – TrackObjects – ParentObjectNumber" (Figure 5.4).

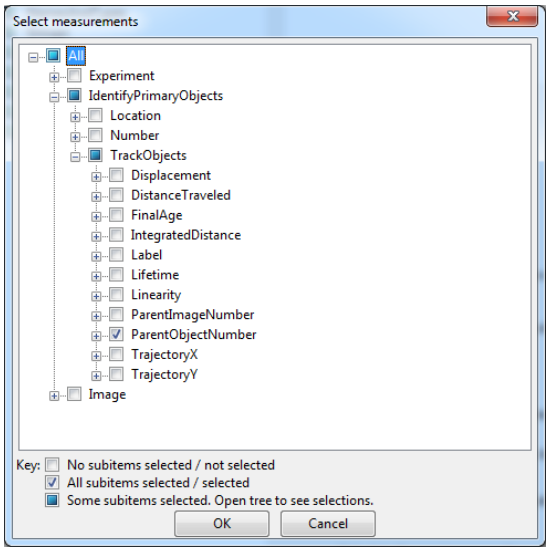



Figure 5.4. Window “Select measurement” in CellProfiler module “ExportToSpreadsheet”.

**Video tutorial**

We made a video showing how to import segmentation and tracking results from CellProfiler. Check video “8 CellProfiler” in the list “eDetect tutorial” at https://www.youtube.com/watch?v=aLqUw92aD8E&list=PLKXmBvkhONktcOXp78igYyeZsCr0wJZg8&index=9.

# 6. Examples

## 6.1. Example 1: live cell imaging data of human HaCaT cell

**Introduction**

In this example, live cell imaging experiments were performed with human HaCaT cells that stably express CFP-H2B nuclear marker and mCherry-Geminin FUCCI cell cycle indicator, whose level indicates that cells are in G1 phase or S/G2/M phases (Sakaue-Sawano A et al., Cell, 2008, PMID: 18267078). Here, we show how to use eDetect to track individual cells and quantify their expression of mCherry-Geminin protein.

**Availability:** This dataset is available at https://github.com/Zi-Lab/eDetect/example/HaCaT-FUCCI.

**Dataset meta-data**

- Cell line: HaCaT_CFP-H2B_mCherry-Geminin
- Channel 1: CFP-H2B (nuclear marker)
- Channel 2: mCherry-Geminin (protein level to be quantified)
- Channel 3: YFP-Smad2 (protein level to be quantified)
- Imaging interval: 15 min per frame

**Filenames**

Channel 1 (left), Channel 2 (middle) and Channel 3 (right) filenames are listed in the table below:

| Channel 1 | Channel 2 | Channel 3 |
|---|---|---|
| HaCaT-FUCCI_t001c1.tif | HaCaT-FUCCI_t001c2.tif | HaCaT-FUCCI_t001c3.tif |
| HaCaT-FUCCI_t002c1.tif | HaCaT-FUCCI_t002c2.tif | HaCaT-FUCCI_t002c3.tif |
| … | … | |
| HaCaT-FUCCI_t206c1.tif | HaCaT-FUCCI_t206c2.tif | HaCaT-FUCCI_t206c3.tif |

In the following sections (**a - i**), we will show how to use eDetect with this example step by step.

A video list "eDetect tutorial" based on this example dataset is available at: https://www.youtube.com/playlist?list=PLKXmBvkhONktcOXp78igYyeZsCr0wJZg8.

Contents of the video list:

1. Starting an analysis: **a - b**
2. Segmentation: **c - d**
3. Tracking: **e - f**
4. Lineage: **g - h**
5. Measurement: **i**
6. Main Interface: Section 3.2 and Section 3.3
7. Scatter plots: Section 4.1
8. Importing CellProfiler results: Section 5.1

### a. Start a New Project

Click on Main Interface. New Project window appears. Click the button Set save path to set the save path of eDetect project file (Figure 6.1).

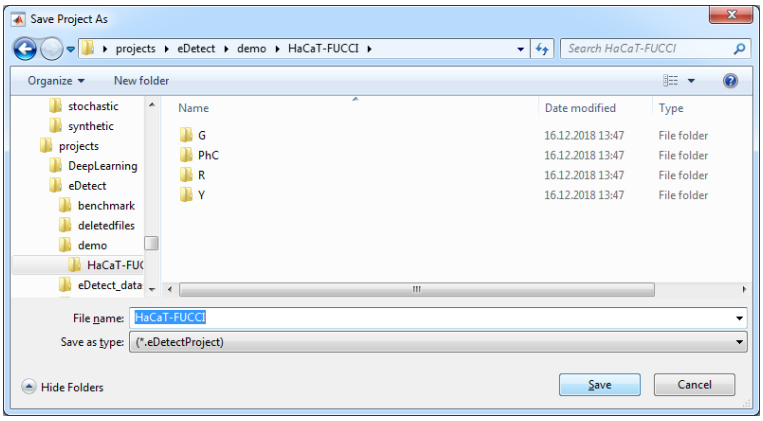



Figure 6.1. Set project file save path

Click "Browse" (to the right of "Channel 1") to set the folder of Channel 1 images (Figure 6.2).

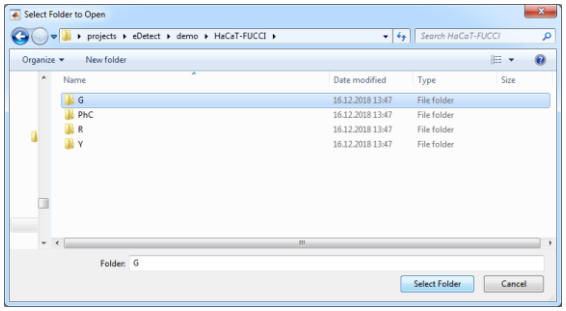



Figure 6.2. Select Channel 1 folder.

Click "Browse" (to the right of "Channel 2") to set the folder of Channel 2 images (Figure 6.3).

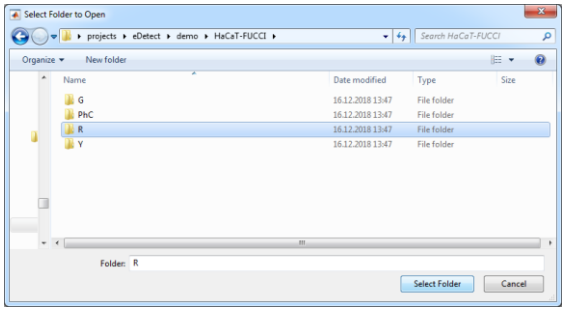



Figure 6.3. Select Channel 2 folder.

Click "Browse" (to the right of "Channel 3") to set the folder of Channel 3 images (Figure 6.4).

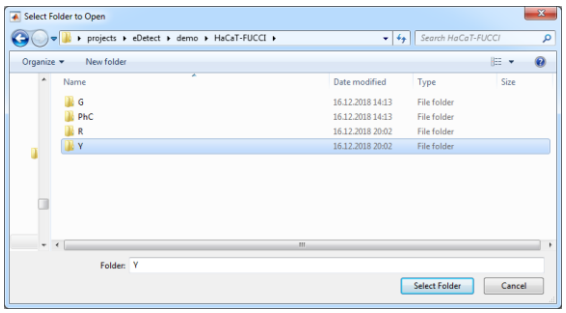



Figure 6.4. Select Channel 3 folder.

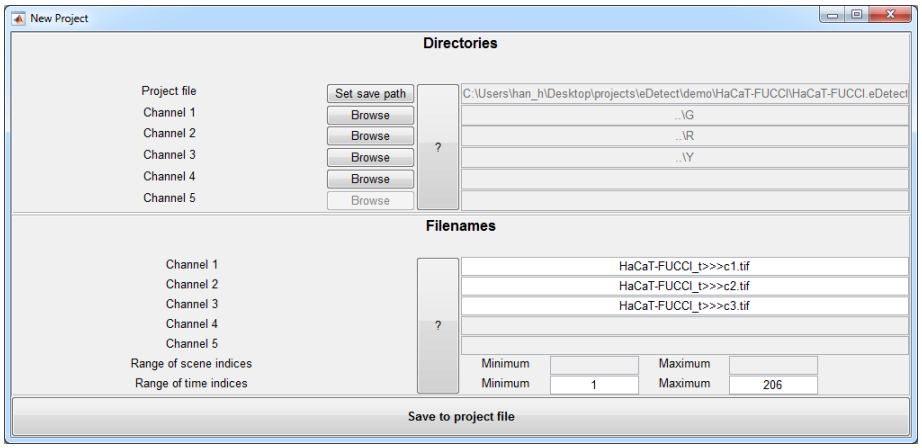



Figure 6.5. Specifying filename formats and ranges of scene and time indices.

Fill out the filename formats and range of time indices (Figure 6.5). Click button Save to project file.

### b. Set Parameters

Click on Main Interface. Directory, Filenames and Parameters window appears. Measure parameters using Measure distance tool in the toolbar on Main Interface. Specify the parameters as shown in Figure 6.6. Click the button Save to project file.

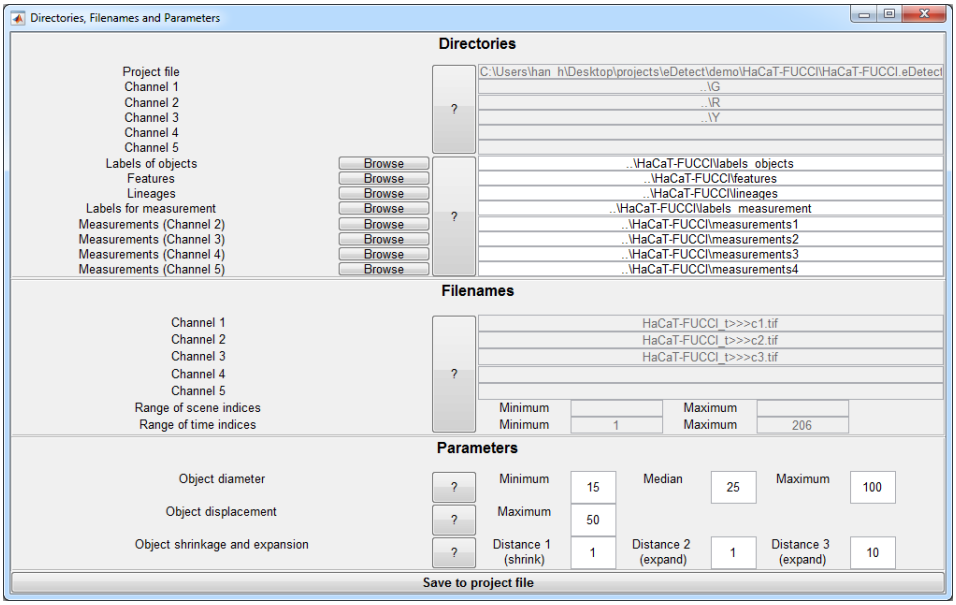



Figure 6.6. Specifying the parameters.

**Note: You can skip steps a and b by loading the eDetect project file (click on Main Interface) "HaCaT-FUCCI.eDetectProject", which has specified the parameters for this example.**

### c. Cell Segmentation and Feature Extraction

Click and wait until Cell Segmentation finishes. Click and wait until Feature Extraction finishes.

### d. Segmentation Gating

After Feature Extraction finishes, the Segmentation Gating window pops up automatically. You can also Click to open the window.

Objects selected (highlighted) in Main Interface are also marked by a red square in Segmentation Gating. The user can select a nucleus to see where it's located in Segmentation Gating (Figure 6.7).

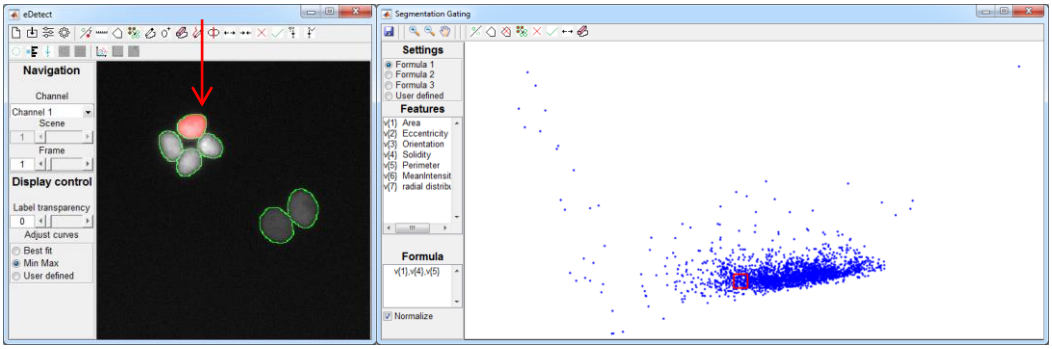



Figure 6.7. Objects selected on Main Interface are marked in Segmentation Gating.

When the user clicks a point in Segmentation Gating, the corresponding object will be highlighted in Main Interface. Therefore, the user can click on points in different regions of the scatter plot to check where the segmentation errors are located in this plot. For example, as shown in Figure 6.8, the outliers (points in the top), the points in the high-density area, and points in the very left represent objects with abnormal shapes (e.g. clumping or overlapping nuclei), correctly segmented nuclei with normal shape and small objects (e.g. over-segmented fragments or debris), respectively.

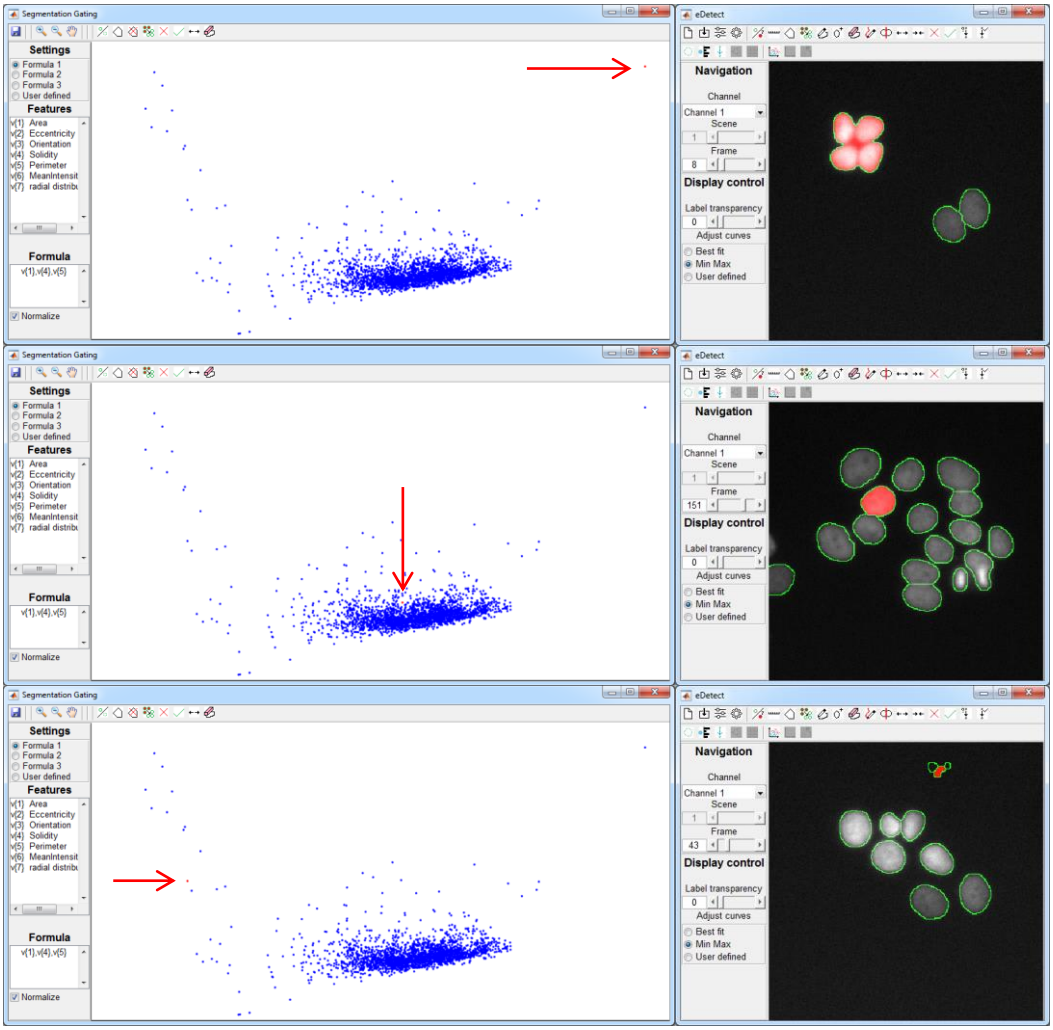



Figure 6.8. Check the shapes of the objects

The user can draw a polygon to select the points in the left, and remove them (Figure 6.9).

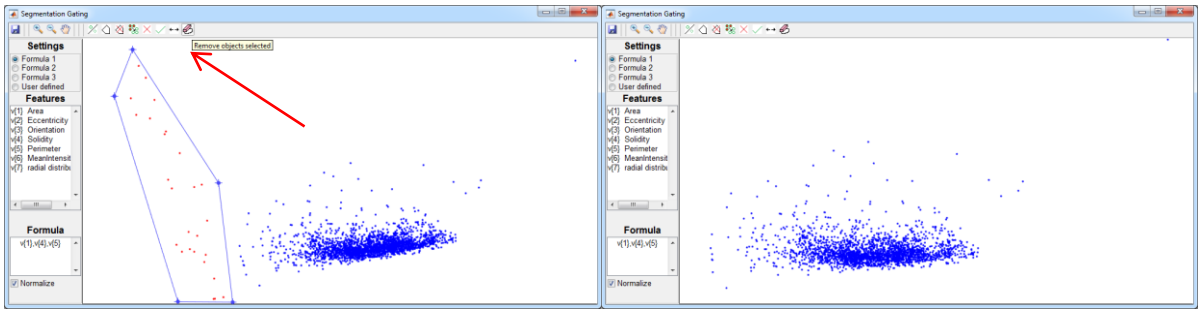



Figure 6.9. Selecting and removing objects in Segmentation Gating.

The object in Figure 6.8 (upper row) is a clump of 4 nuclei and is probably difficult to split automatically. Therefore we need to click on the tool “Draw division”, draw division lines, and divide the object into pieces by clicking “Divide objects” (Figure 6.10). Afterwards we select the background area in the middle of the 4 nuclei and click on “Remove objects”.

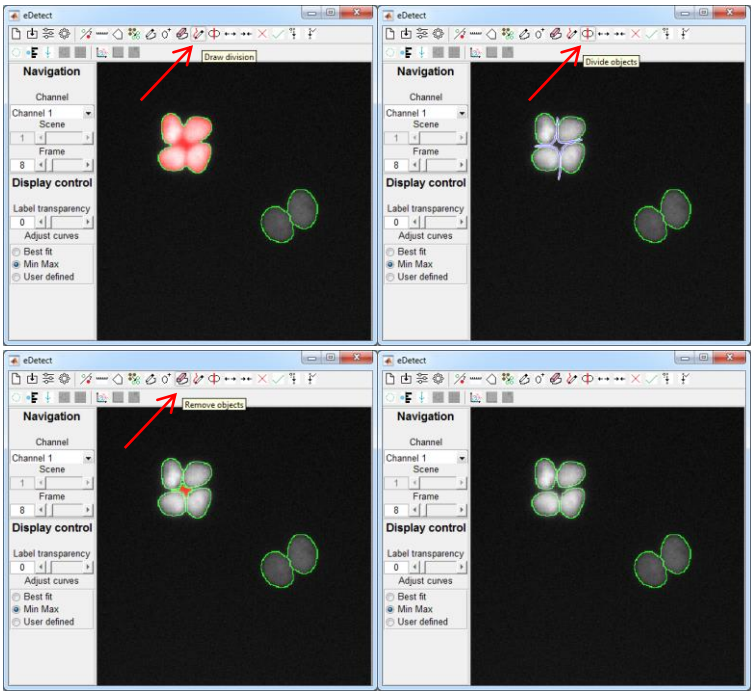



Figure 6.10. Manually divide a clump of 4 nuclei.

### e. Cell Tracking

Click  and wait until Cell Tracking finishes.

### f. Cell Pair Gating

After Cell Tracking finishes, the Cell Pair Gating window pops up automatically. You can also Click  to open the window. Similar to Segmentation Gating, the user can click on a point in Cell Pair Gating to see which two objects share the same predecessor in the previous frame. If they are really a pair of newborn daughter cells, then we keep them. Otherwise, we correct the error that caused this fake division.

**An example of a correct pair of sister cells**

Figure 6.11 shows a pair of newly divided daughter cells.

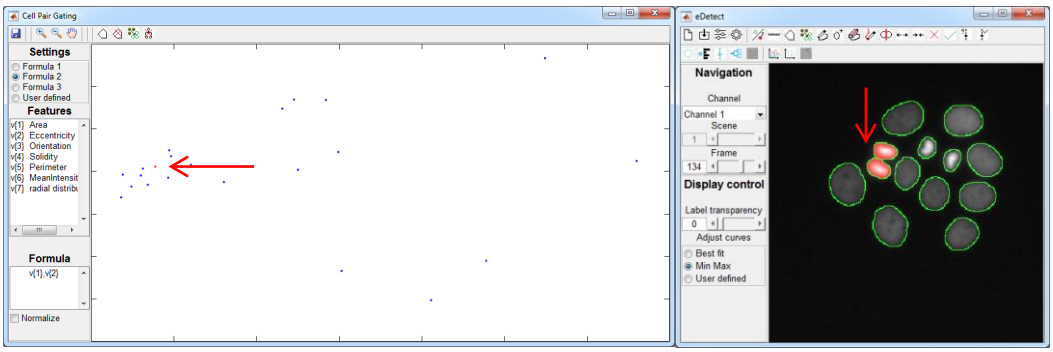



Figure 6.11. A pair of newly divided daughter cells.

**An example of segmentation error caused by over-segmentation**

In Figure 6.12, the pair of objects that share a common predecessor are actually two parts of one nucleus (panel b). Both of them are already selected, we only need to click Merge objects (panel b). Then two objects become one (panel c), and the point representing the pair of objects is removed (panel d).

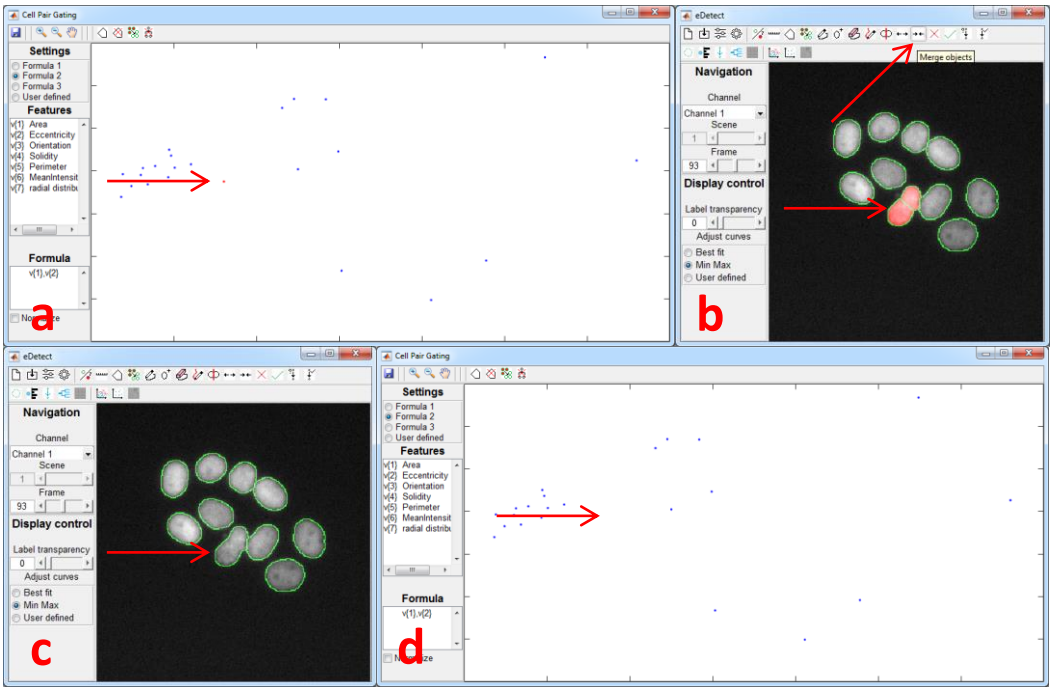



Figure 6.12. Detecting and correcting a segmentation error (over-segmentation).

**An example of segmentation error caused by nuclei clumping**

In Figure 6.13, we click on a point (panel a) and two objects are selected on Main Interface (panel b). We select one of them (by deselecting the other) and click Get predecessor (panel c). The predecessor turns out to be a clump of 2 nuclei (panel d). We click on "Split objects" (panel d) but it doesn't work (panel e). We click on "Draw division" (panel e), draw lines and click "Divide objects" (panel f). The object is divided into 2 (panel g) and Cell Pair Gating panel is updated (panel h).

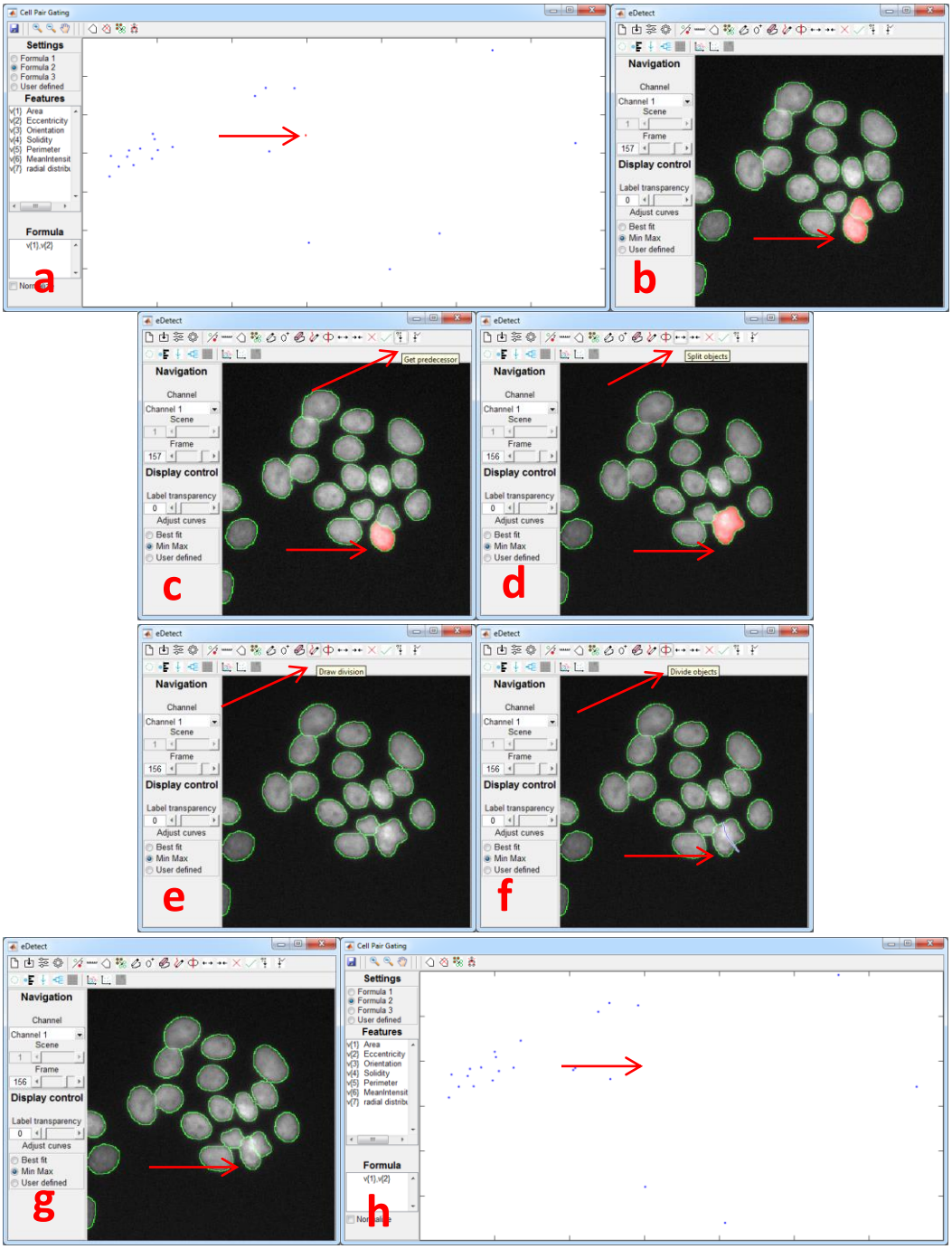



Figure 6.13. Detect and correct a segmentation error caused by nuclei clumping.

**Another example of segmentation error caused by nuclei clumping**

In Figure 6.14, we click on a point (panel a) and two objects are selected on Main Interface (panel b). We select one of them (by deselecting the other) and click Get predecessor (panel c). The predecessor turns out to be a clump of 2 nuclei (panel d). We click on “Split objects” (panel d) and the object is divided into 2 (panel e) and Cell Pair Gating panel is updated (panel f).

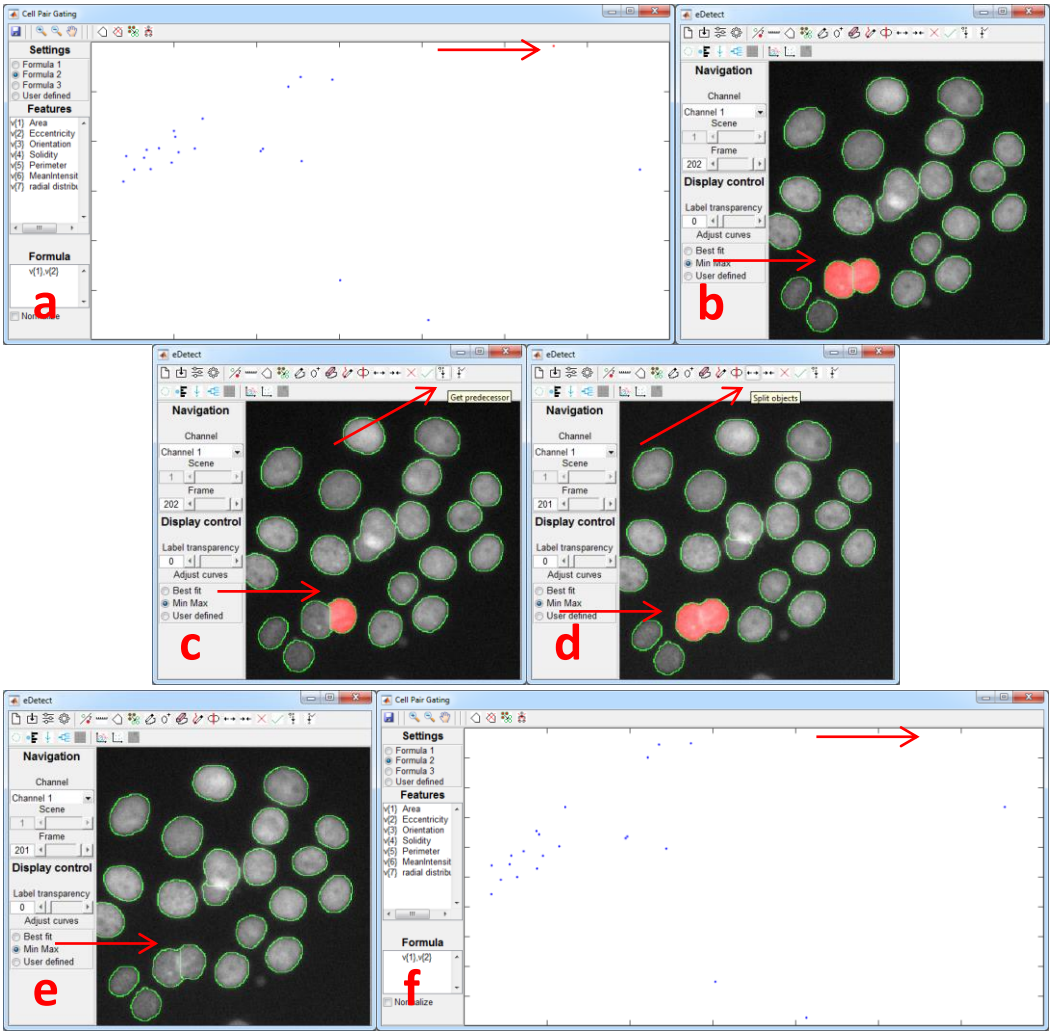



Figure 6.14. Detecting and correcting a segmentation error caused by nuclei clumping.

**An example for correcting a tracking error caused by cell division**

Figure 6.15 shows an example of cell tracking error caused by cell division. Apparently cell 41.1 and cell 41.2 are not a pair of newly divided daughter cells (panel b). Instead, cell 41.1 and cell 41.3 are (panel b). We can select cell 41.1 (deselect cell 41.2) and click Get predecessor (panel c). The automatically assigned predecessor is cell 40.1 (panel d), which is wrong. The correct predecessor of cell 41.1 (panel c) is actually cell 40.2 (panel d). This tracking error occurs because during the cell division, one daughter cell (41.1 in panel c) moves by such a large distance that it is closer to 40.1 than to 40.2 (panel d). To correct the error we need to deselect cell 40.1 and select cell 40.2 (panel e). Then we click Set predecessor to make the change (panel e). Now the point representing the pair is removed (panel f).

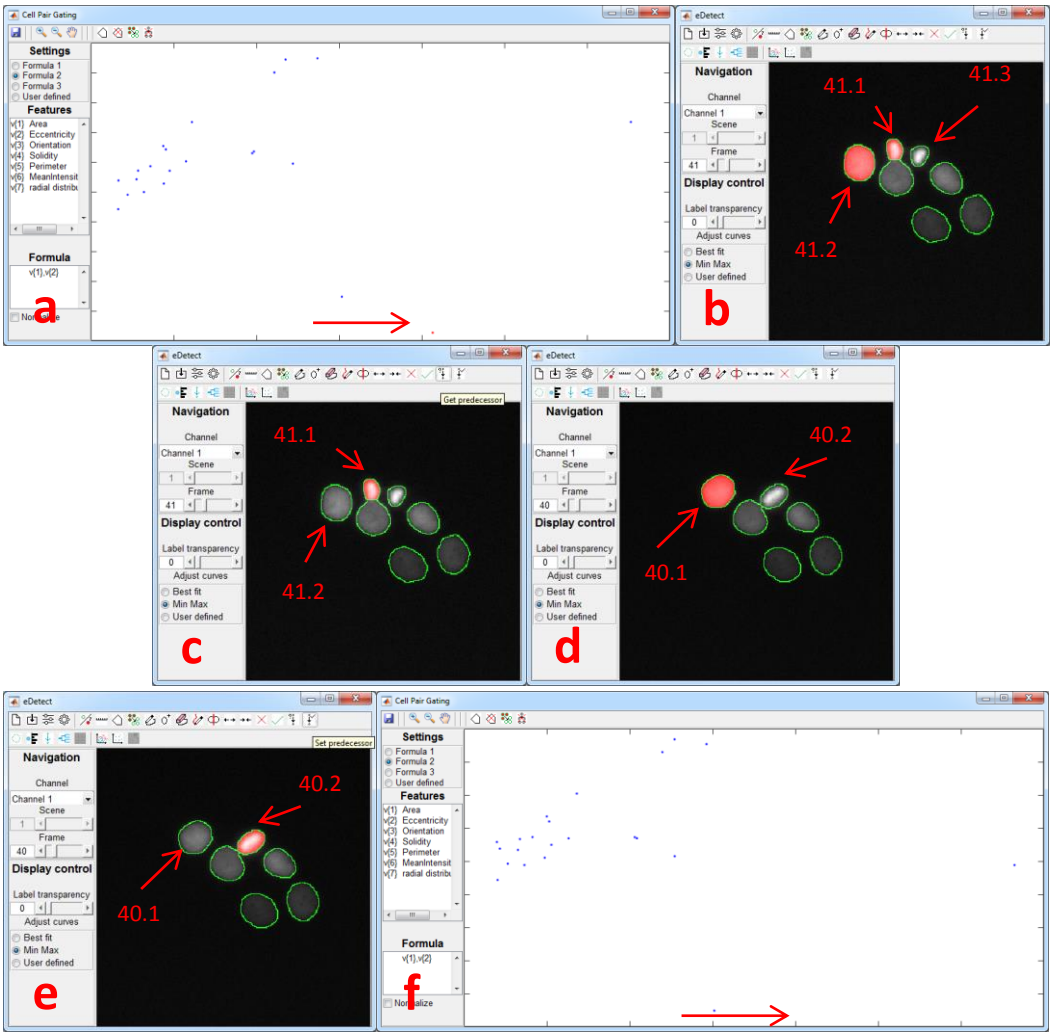



Figure 6.15. Detecting and correcting a tracking error caused by cell division.

**An example for correcting a tracking error caused by fast cell migration**

Figure 6.16 illustrates another example, in which the selected 2 nuclei (cell 3.1 and cell 3.2 in panel b) have normal nuclear size and they don't look like newly divided daughter cells. The user can select cell 3.2 (deselect cell 3.1) and click Get predecessor (panel c). The automatically assigned predecessor (cell 2.1 in panel d) is incorrect. The real predecessor should be cell 2.2 (panel d). The user can deselect the wrong predecessor (cell 2.1 in panel e) and select the correct one (cell 2.2 in panel e). Then we click Set predecessor to make the change (panel e). Now the point representing the pair is removed (panel f).

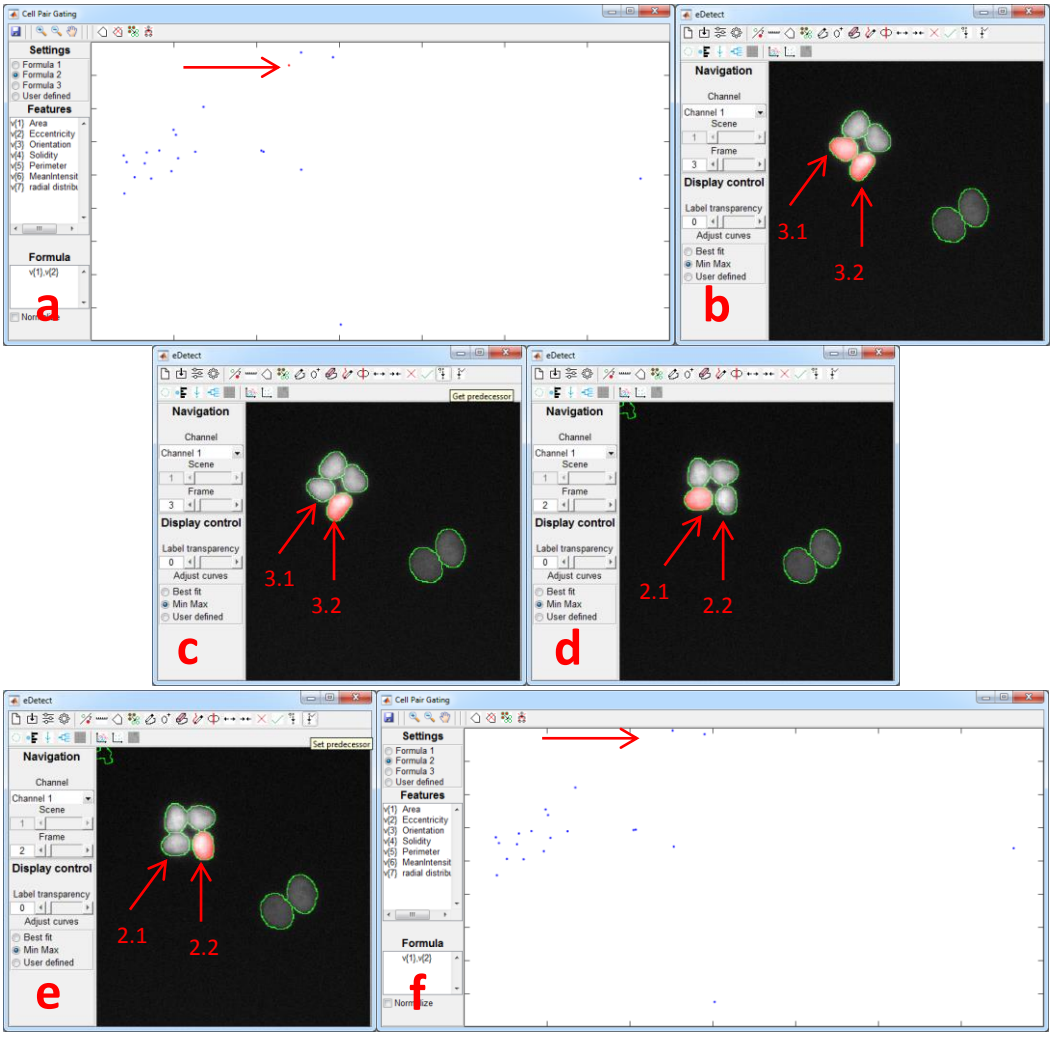



Figure 6.16. Detect and correct a tracking error caused by fast cell migration.

**Note**

We should keep checking each point until there are no more fake cell divisions. Then we proceed to the next quality control module "Cell Lineages Display" to see whether there are other errors.

### g. Cell Lineage Reconstruction

Click and wait until Cell Lineage Reconstruction finishes.

### h. Cell Lineages Display

After Cell Lineage Reconstruction finishes, the Cell Lineages Display window pops up automatically. The user can also click  to open the window (Figure 6.17).

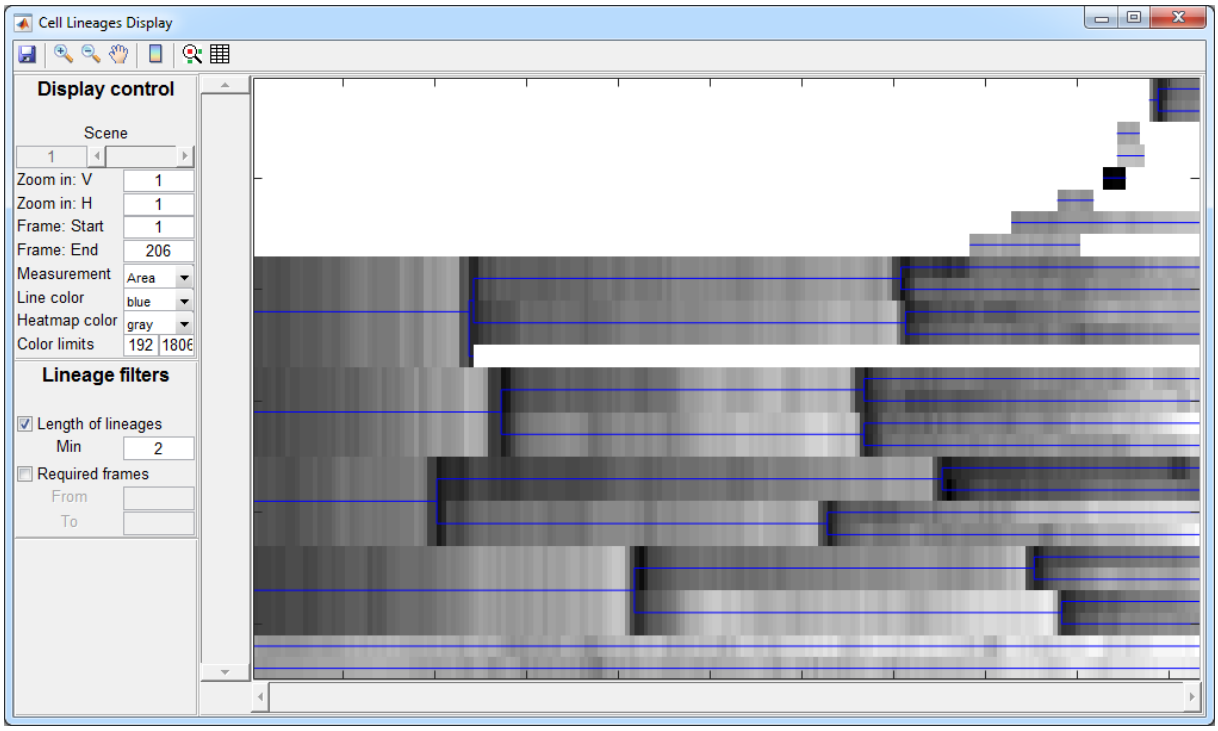



Figure 6.17. Cell Lineages Display.

The user can click  to detect outliers, which will be highlighted in red (Figure 6.18).

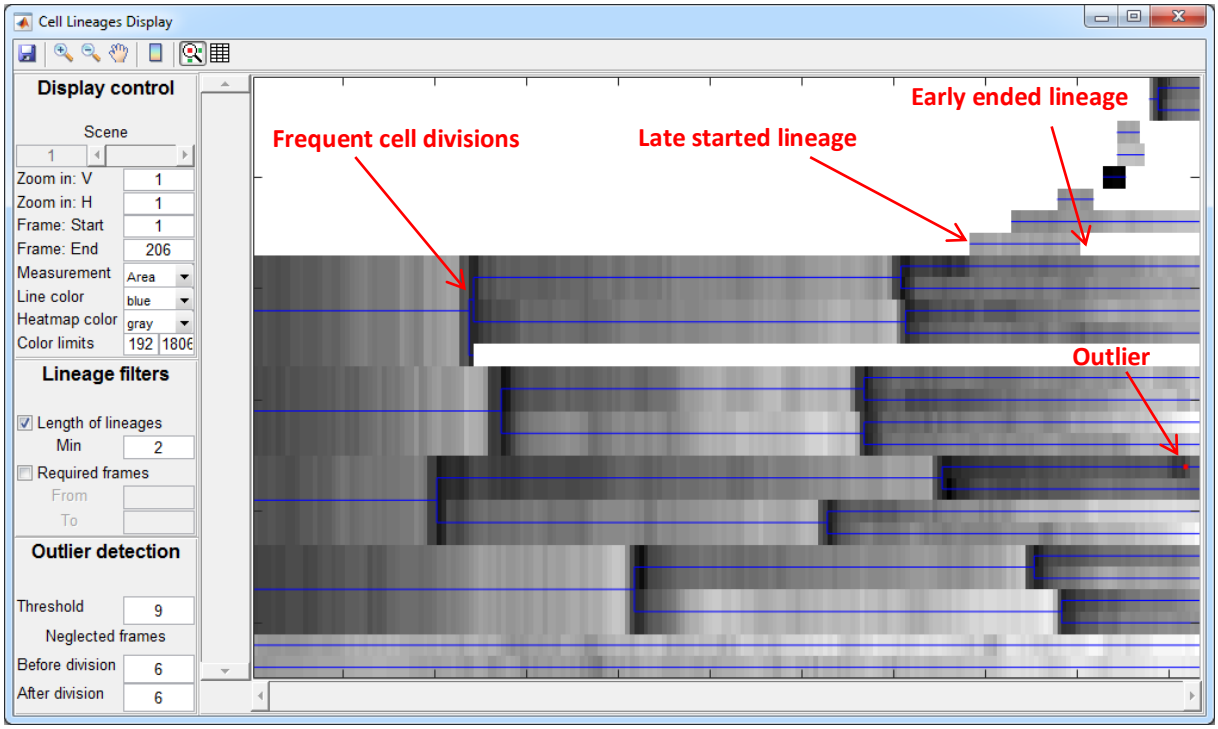



Figure 6.18. Cell Lineages Display with outlier detection function on.

As mentioned in Section 4.3, we advise users to pay attention to 4 types of patterns for further error check: frequent cell divisions (Figure 6.19), early ended lineages (Figure 6.20), late started lineages (Figure 6.21) and outliers (Figure 6.22).

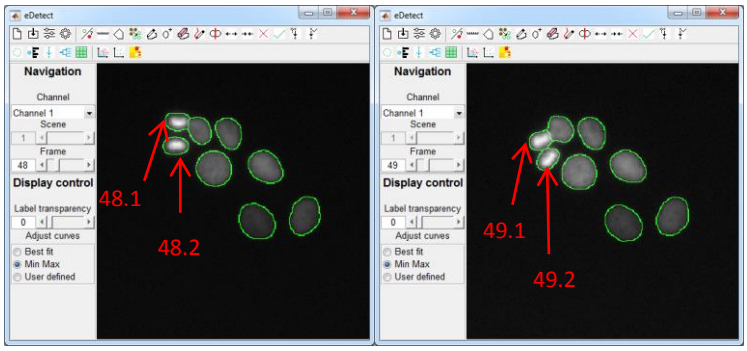



Figure 6.19. An example of frequent cell divisions.

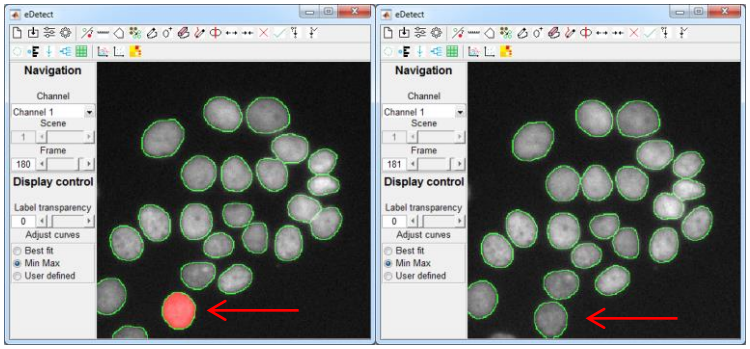


Figure 6.20. An example of early ended lineages.

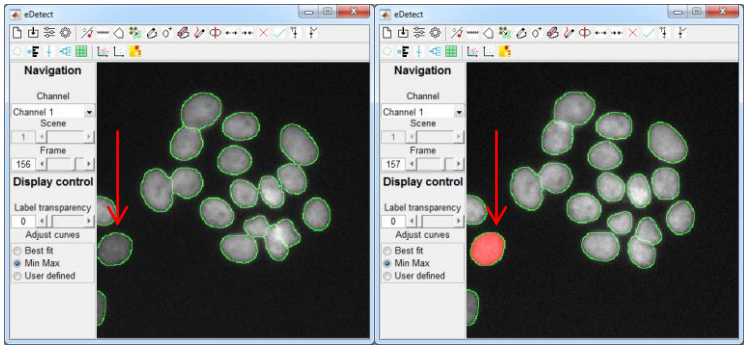


Figure 6.21. An example of late started lineages.

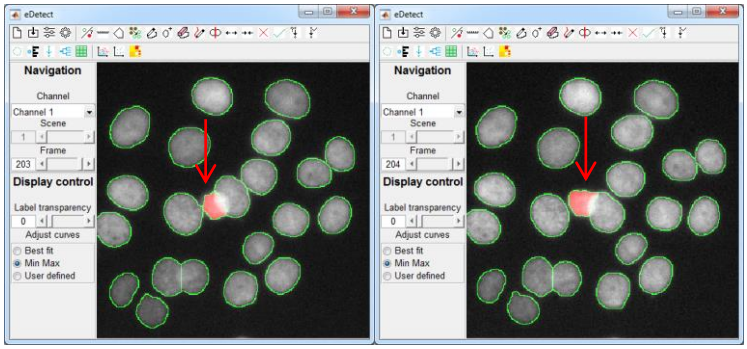


Figure 6.22. An example of outliers.

In Figure 6.19, the second “division” is not really a cell division. Both 49.1’s and 49.2’s automatically assigned predecessors are 48.2, but 49.1’s predecessor should be 48.1. The object 49.1 is closer to 48.2 than to 48.1 because it moved a large distance between the 2 frames. In Figure 6.20, the lineage ended early because the cell started touching the border at Frame 181 (cells touching image border are excluded from Cell Tracking). In Figure 6.21, the lineage started late because the cell enters the field late and it stops touching the border at Frame 157. In Figure 6.22, the sudden change in object area is caused by inaccurate segmentation caused by nuclei overlapping and heterogeneous brightness of the nuclei, both of which are because the highlighted nucleus is deeper (along z-axis) than others.

After correcting the detected errors, the user can tick Required frames in Lineage filters to filter out the incomplete lineages (Figure 6.23).

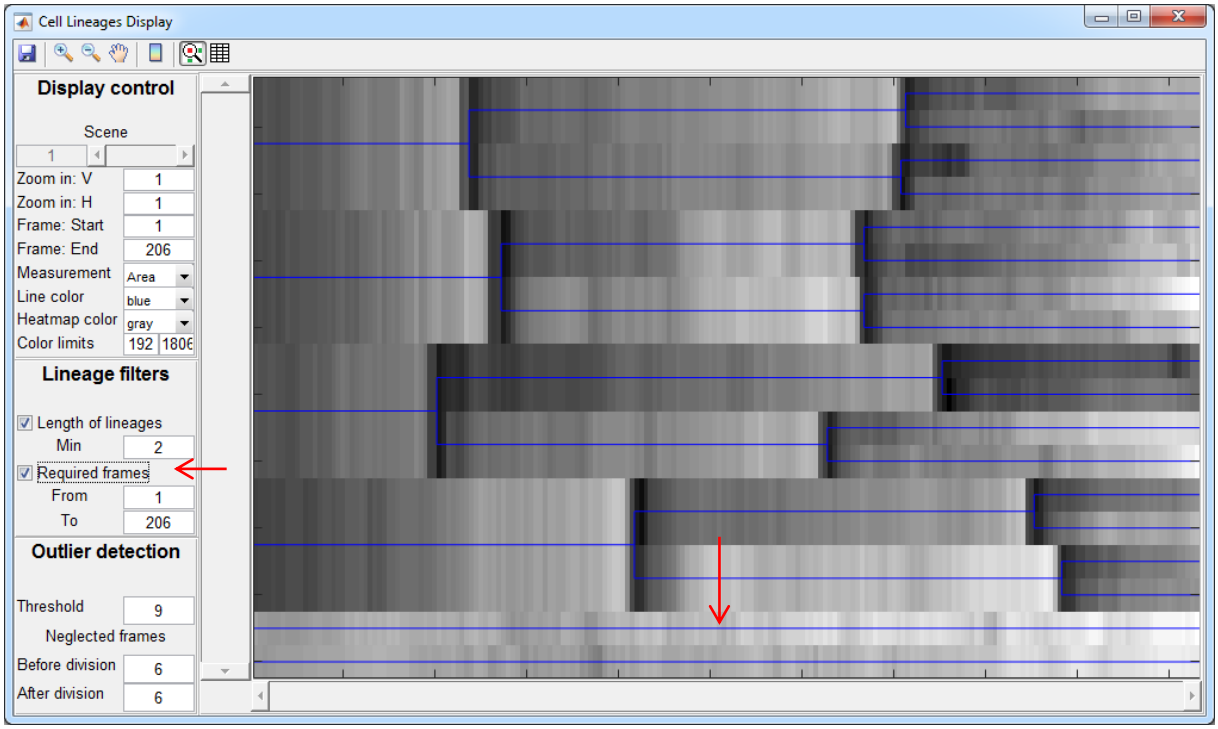



Figure 6.23. Cell Lineages Display. Incomplete lineages are not displayed.

If you right-click on one of the lineages (Figure 6.23), a Synchrogram window (Figure 6.24) appears.

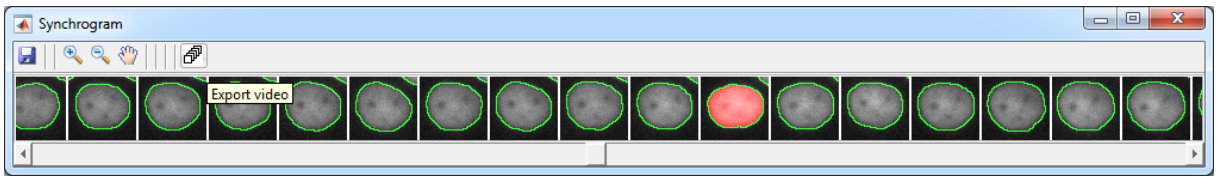



Figure 6.24. Synchrogram of the seleted cell lineage.

The user can Export video on Synchrogram (Figure 6.24). A series of cropped image of the cell lineage are stored in the folder where the lineage files are stored (Figure 6.25).

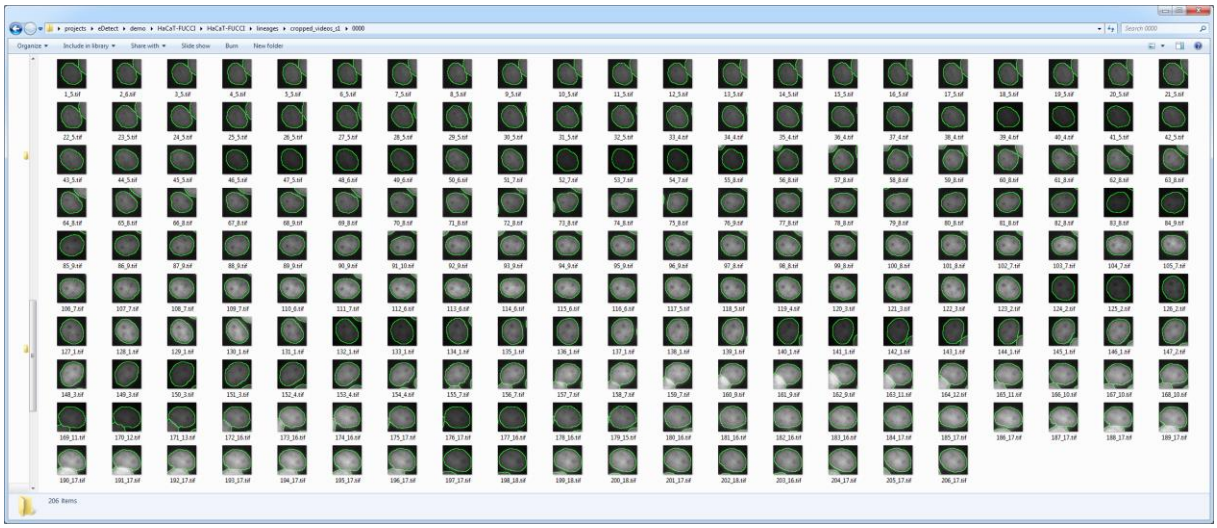


Figure 6.25. Images exported from Synchrogram.

In addition, the user can change the visualized Measurement from Area to MeanIntensity (Figure 6.26).

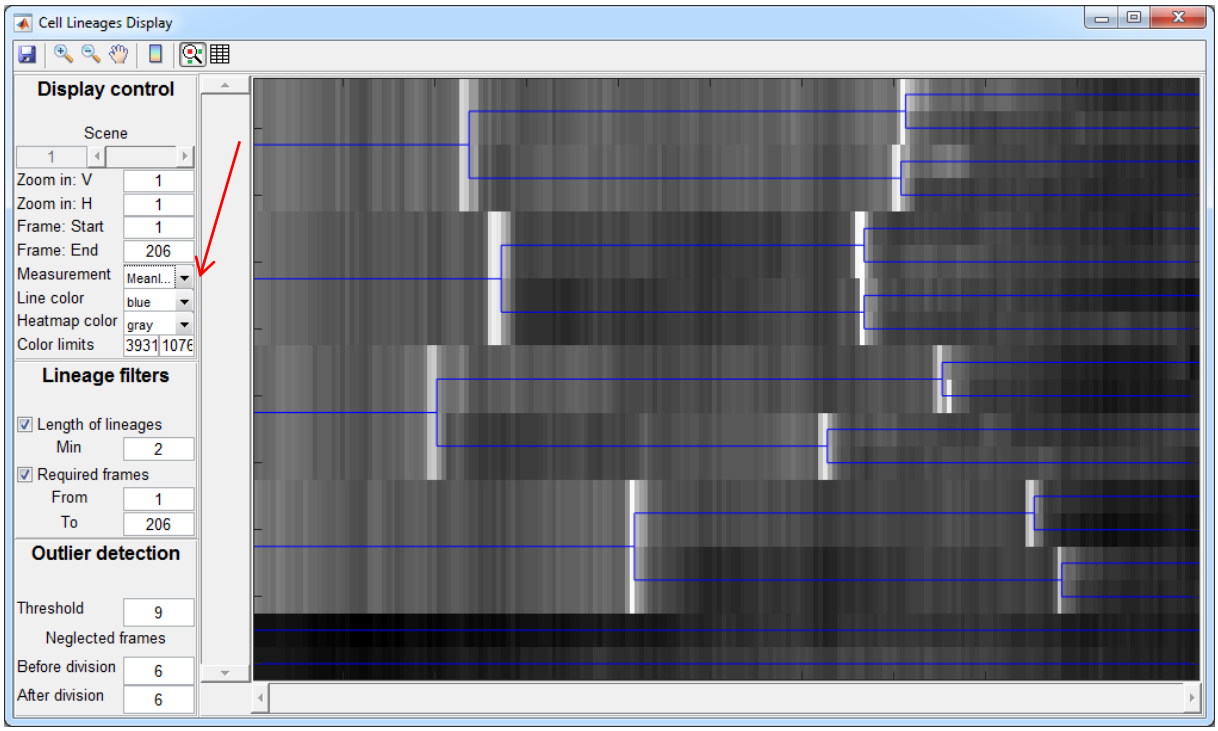



Figure 6.26. Cell Lineages Display.

### i. Measurement

Click and wait until Measurement finishes. In Cell Lineages Display, the intensity statistics of Channel 2 (mCherry-Geminin protein (FUCCI signal)) and Channel 3 (YFP-Smad2) are also available for display (Figure 6.27).

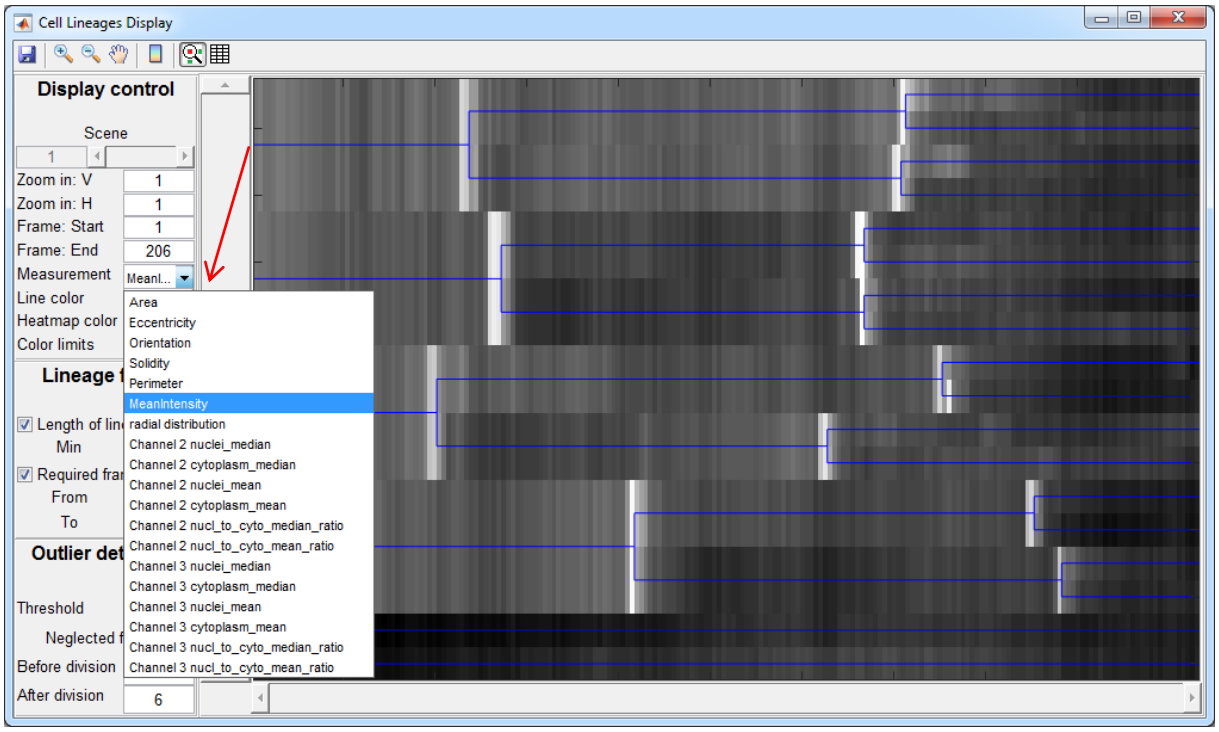



Figure 6.27. Cell Lineages Display.

We select “Channel 2 nuclei_median” and turn on the colorbar (Figure 6.28).

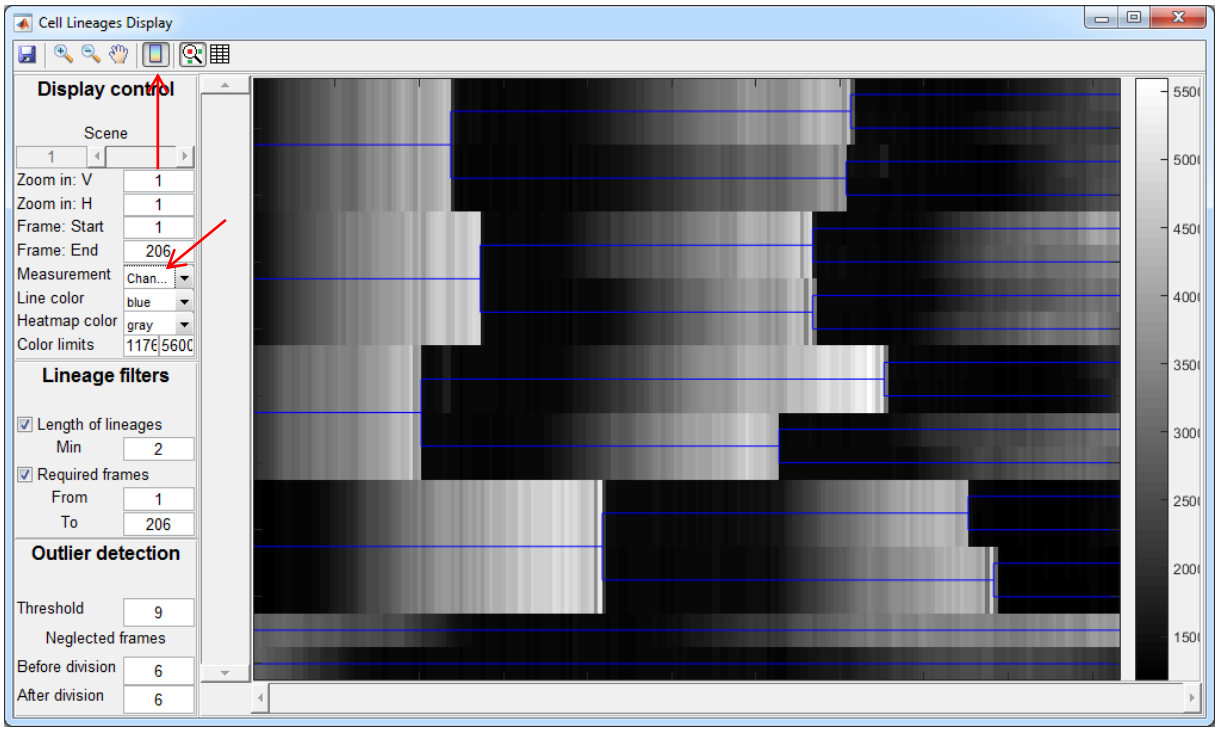



Figure 6.28. Cell Lineages Display.

Other colormaps are also available. For example in Figure 6.29, “hot” is used and colorbar is shown.

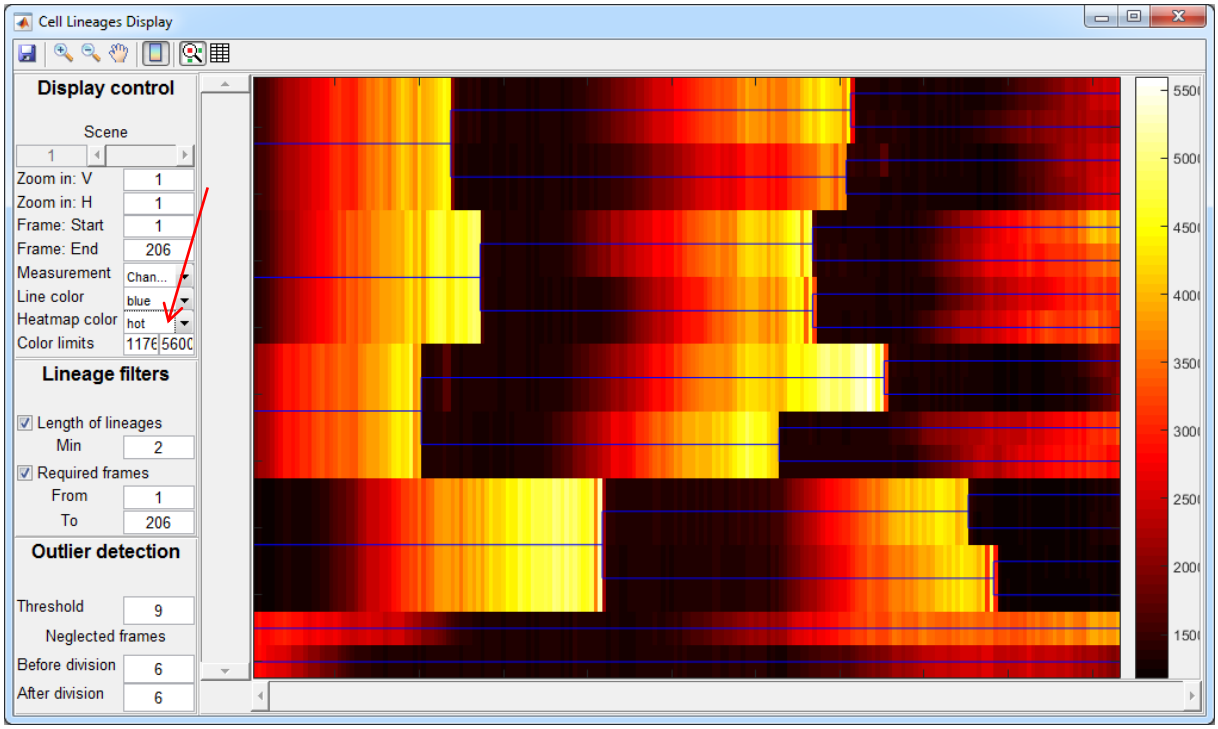



Figure 6.29. The dynamics of mCherry-Geminin protein in HaCaT single cells.

The user is able to save the data by clicking Export table Figure 6.30 shows the dynamics of mCherry-Geminin protein (FUCCI signal) in the complete lineages.

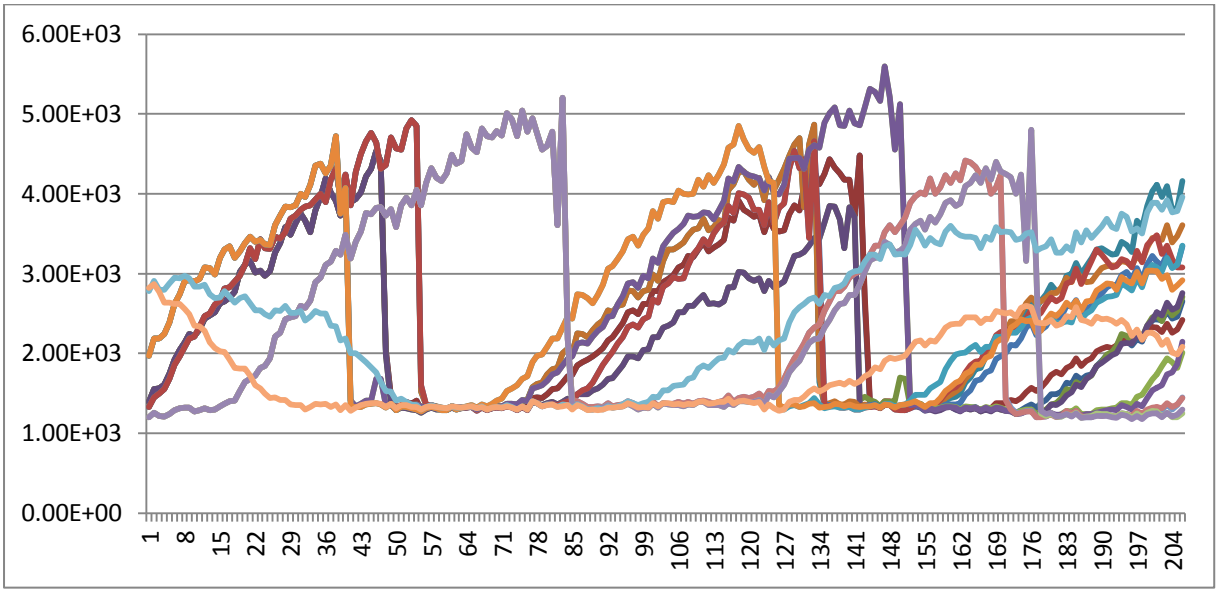



Figure 6.30. The dynamics of mCherry-Geminin protein in HaCaT single cells.

## 6.2. Example 2: live cell imaging data of mouse stem cells

**Introduction**

This section shows the results of processing the live-cell imaging data from mouse stem cell, which is described at Cell Tracking Challenge (http://www.celltrackingchallenge.net/). It is available for download at http://data.celltrackingchallenge.net/training-datasets/Fluo-N2DH-GOWT1.zip after registration. The video "01" of this dataset are used for demonstration. Here, we show how to use the Segmentation Gating module to detect segmentation errors.

**Start the project**

Download the eDetect project file "Fluo-N2DH-GOWT1_01_project.eDetectProject" from "https://github.com/Zi-Lab/eDetect/example/Fluo-N2DH-GOWT1". Put the eDetect project file in the folder "Fluo-N2DH-GOWT1" together with the folders "01" and "02". Start eDetect and load the eDetect project file.

**Cell Segmentation and Feature Extraction**

Click and wait until Cell Segmentation finishes. Click and wait until Feature Extraction finishes.

**Segmentation Gating**

After Feature Extraction finishes, the Segmentation Gating window pops up automatically. It can also be opened by clicking .

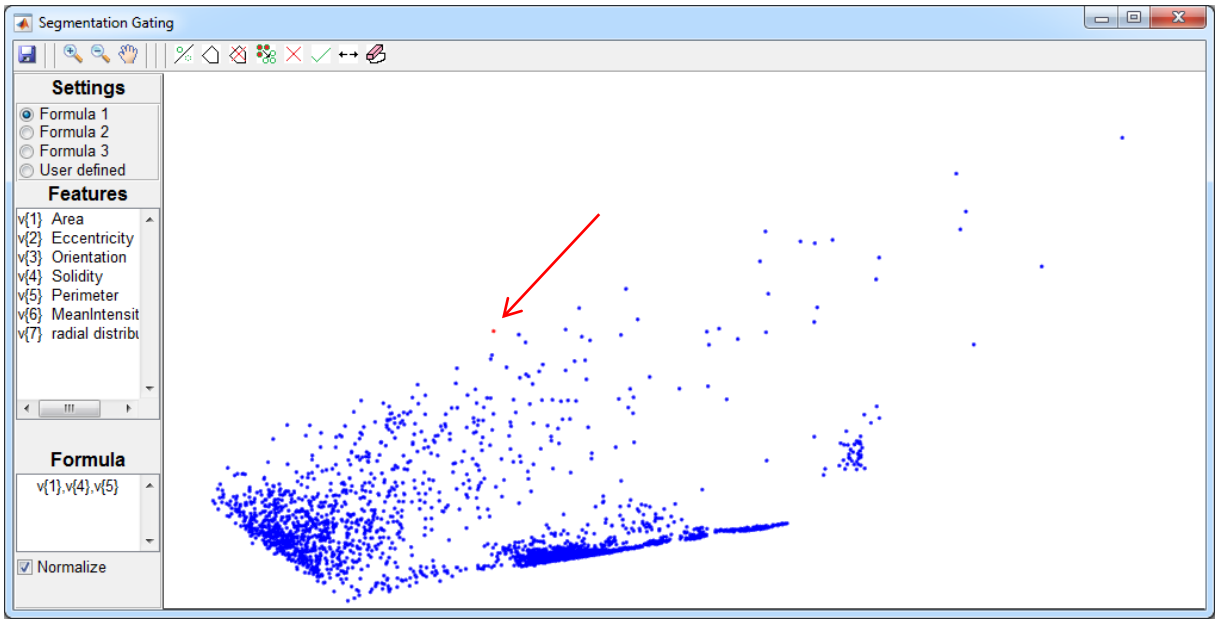



Figure 6.31. Click an outlier point in the Segmentation Gating window

If we click on a point as shown in Figure 6.31, it corresponds to a bright background region (Figure 6.32).

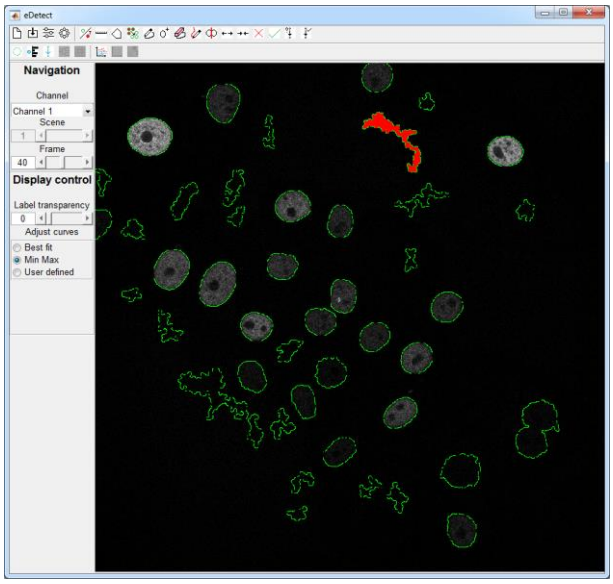



Figure 6.32. The clicked point corresponds a bright background region.

After clicking on several other points, the user can conclude that the points in the upper part are mostly bright background regions. We can draw a polygon to select and delete them (Figure 6.33).

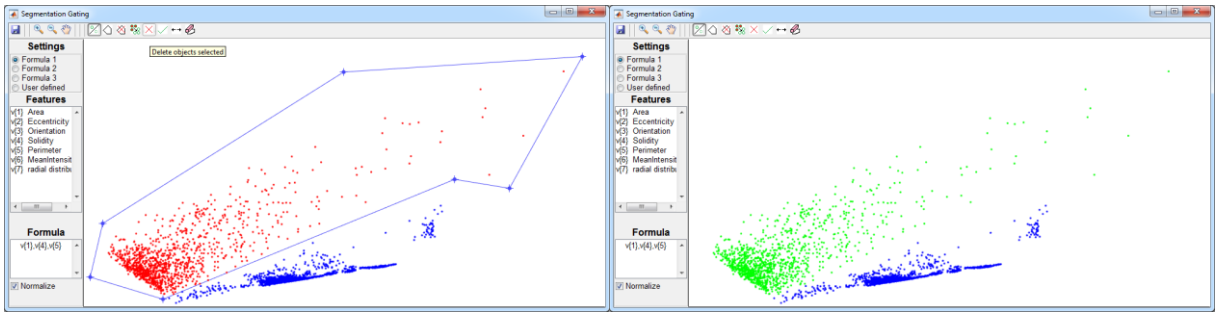



Figure 6.33. Select and delete.

Now we click on the group of points in the right (Figure 6.34).

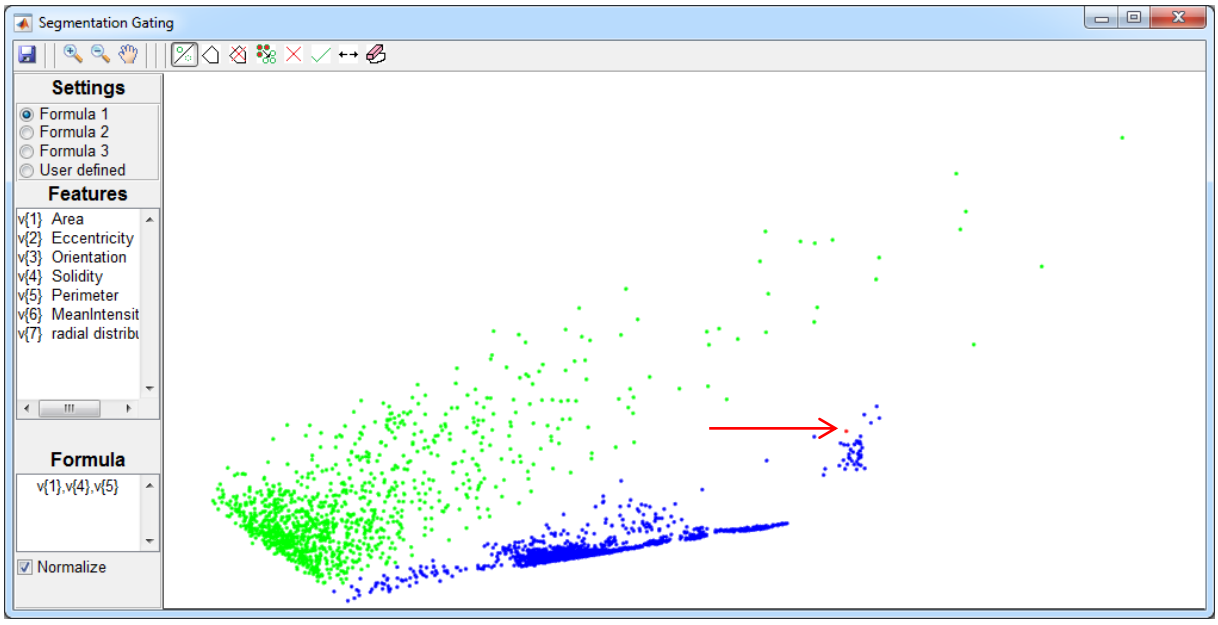



Figure 6.34. Clicking on a point in a cluster.

The cluster of points in the right (Figure 6.34) correspond to nuclei clumps (Figure 6.35).

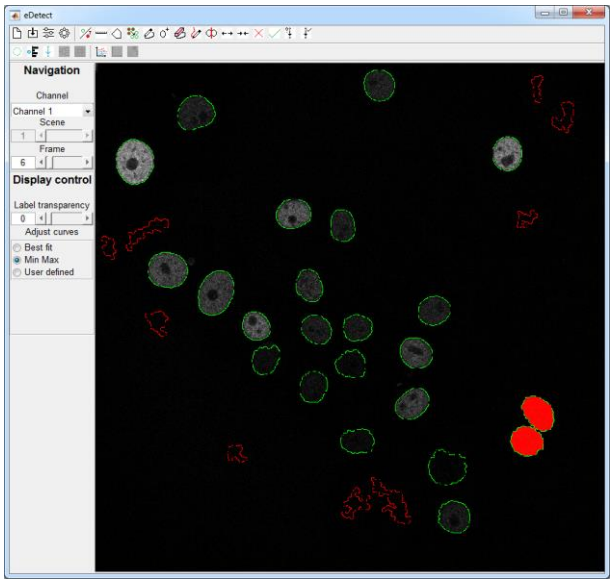



Figure 6.35. The clicked object is a clump of 2 nuclei.

The user can draw a polygon to select and split all of them (Figure 6.36).

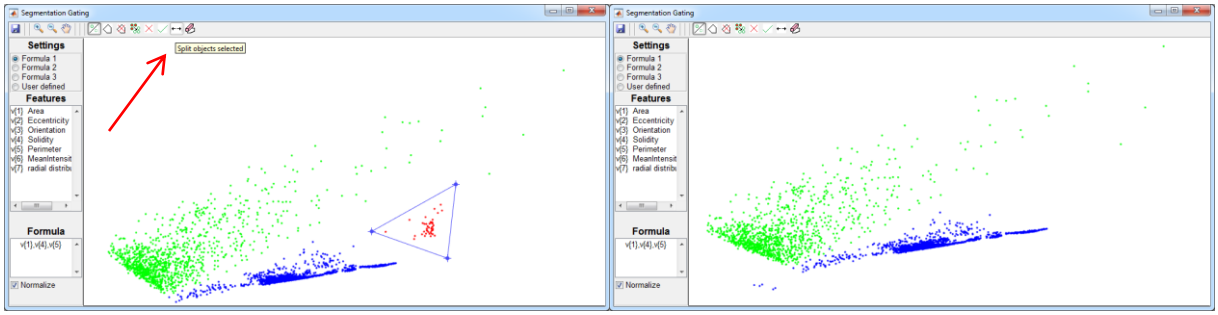



Figure 6.36. Selecting and splitting nuclei-clumps.

# 7. Troubleshooting

If you have problems running eDetect, please try solving them with the help of this note. If problems still exist, please feel free to contact us (see Section 8.3 Support).

- **Problem: Cell segmentation is very slow.**

**Solution 1:** In Settings: Segmentation settings change "De-clumping and merging" from "With de-clumping and merging" to "Without de-clumping or merging". And then, after Cell Segmentation you can de-clump manually on Main Interface or Segmentation Gating.

**Solution 2:** In Settings: Segmentation settings change "Maximum number of objects to merge" from 3 to 2. Maybe there will be more segmentation errors (depending on the dataset) but the process will probably become faster.

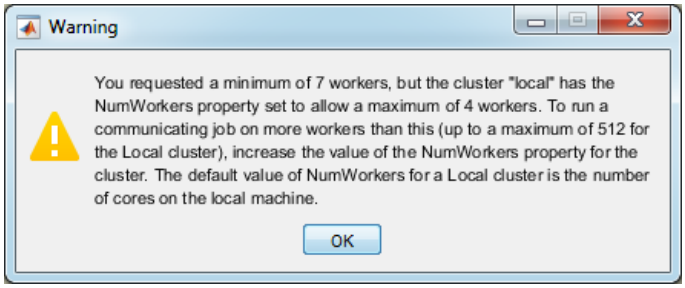
**Solution 3:** In Settings: Segmentation settings change "Maximum time for a connected component" from 100 to lower values. Maybe there will be more segmentation errors (depending on the dataset) but the process will probably become faster.

- **Problem: manual correction is very slow.**

**Solution:** In Settings: Manual correction settings change "After each manual correction of segmentation update" from "Object labels, features, tracks and lineages" to other options.

- **Problem: MATLAB warning on number of workers (while running eDetect source code).**

See Figure below.




**Solution:** Go to Preferences → Parallel Computing Toolbox → Cluster Profile Manager → Number of workers to start on your local machine (NumWorkers). Put a number that is not smaller than the number in Settings → Other settings → Number of processors to use.

# 8. About

## 8.1. License

The project is licensed under the MIT license.

## 8.2. Third party codes and datasets references

- CellProfiler 1.0
  https://github.com/CellProfiler/CellProfiler-1.0
- Parfor_progressbar (Daniel Terry)
  https://www.mathworks.com/matlabcentral/fileexchange/53773-parfor-progressbar
- Waitbar with time estimation (Andrew)
  https://www.mathworks.com/matlabcentral/fileexchange/22161-waitbar-with-time-estimation
- Minimum Volume Enclosing Ellipsoid (Nima Moshtagh)
  https://www.mathworks.com/matlabcentral/fileexchange/9542-minimum-volume-enclosing-ellipsoid
- Relative path (Jochen Lenz and Dan O'Shea)
  https://www.mathworks.com/matlabcentral/fileexchange/3858-relativepath-m
  https://github.com/djoshea/matlab-utils/blob/master/path/relativepath.m
- Cell Tracking Challenge
  http://www.celltrackingchallenge.net/

## 8.3. Support

If you have any questions or suggestions, please feel free to contact us via email:
han_h@molgen.mpg.de (Hongqing Han) or zhike.zi@molgen.mpg.de (Dr. Zhike Zi).