HHS Public Access

Author manuscript

Methods Cell Biol. Author manuscript; available in PMC 2015 July 16.

Published in final edited form as:

Methods Cell Biol. 2014 ; 123: 409–427. doi:10.1016/B978-0-12-420138-5.00022-7.

USER-FRIENDLY TOOLS FOR QUANTIFYING THE DYNAMICS OF CELLULAR MORPHOLOGY AND INTRACELLULAR PROTEIN CLUSTERS

Denis Tsygankov[1], **Pei-Hsuan Chu**[1], **Hsin Chen**[2], **Timothy C. Elston**[1], and **Klaus Hahn**[1]

[1]Department of Pharmacology and Lineberger Cancer Center, University of North Carolina, Chapel Hill, NC 27599, USA

[2]Department of Pharmacology and Cancer Biology, Duke University, Durham, NC 27710, USA

Abstract

Understanding the heterogeneous dynamics of cellular processes requires not only tools to visualize molecular behavior, but also versatile approaches to extract and analyze the information contained in live-cell movies of many cells. Automated identification and tracking of cellular features enables thorough and consistent comparative analyses in a high-throughput manner. Here we present tools for two challenging problems in computational image analysis: 1) classification of motion for cells with complex shapes and dynamics, and 2) segmentation of clustered cells and quantification of intracellular protein distributions based on a *single* fluorescence channel. We describe these methods, and user-friendly software* (MATLAB applications with Graphical User Interfaces) so these tools can be readily applied without an extensive knowledge of computational techniques.

Keywords

image quantification; motion classification; cell segmentation; cell tracking; user interface

Introduction

Quantitative image analysis not only provides rigor in the comparison of experimental observations, but also enables the extraction of information that is not apparent to the naked eye(Danuser, 2011). This is particularly true for correlation analyses, given that biological data often exhibit high variability. Automated microscopes provide the capabilities to image large populations of cells on a cell by cell basis, enabling correlative studies despite heterogeneous responses across cell populations, but these capabilities also present challenges for image analysis methods, as the heterogeneous responses must be imaged and characterized with minimal human intervention.

One example of such a challenge is the quantification of changes in the morphology of cells with complex geometries. Methods have been developed to characterize and classify static

*https//dl.dropboxusercontent.com/u/7035514/GUIs_for_MCB.zip

cell shapes(Loncaric, 1998; Sailem, Bousgouni, Cooper, & Bakal, 2014; Zhang & Lu, 2004), but modern live cell microscopy requires us to address the quantitation and analysis of cell *dynamics*. Studying motility, immune synapse formation, tissue development and a host of other processes requires understanding the evolution of complex shapes(Machacek et al., 2009). Here we present a method for automated classification of cell motion types, and a user-friendly tool based on this method. The method differentiates between six types of cellular motion and assigns one of the motion types to each time frame in a movie. The parameters of the method can be interactively adjusted in a GUI (graphical user interface) provided with the software. By applying this tool to many cells one can establish the dominant modes of motion at different stages of transient cell dynamics and quantify the effects of drugs or mutations on cell morphology and motility.

Another image analysis challenge involves the segmentation of tightly packed cells, which is necessary for automated quantification of many cell behaviors. Previous studies addressed this challenge with the help of fluorescent nuclear or membrane markers [see (Doncic, Eser, Atay, & Skotheim, 2013; Indhumathi, Cai, Guan, & Opas, 2011; Malpica et al., 1997; Nilsson & Heyden, 2005; Schmitt & Reetz, 2009) and references therein]. However, cell behaviors of interest must frequently be detected at the same time using fluorescent probes (e.g. to track intracellular localization of proteins or movements of subcellular structures); and additional markers for segmentation may obscure the behavior of interest. In addition, fluorescence microscopy can induce significant phototoxicity(Carlton et al., 2010), and imaging larger numbers of fluorescence probes can limit either the duration of filming or the frequency with which images can be captured without undue harm to the cells. To circumvent these issues, we developed a method for *simultaneous* segmentation of tightly packed cells and quantification of protein clusters based on a *single* fluorescent marker. The method is ideal for cells with a simple geometry (such as yeast), and has the flexibility to address several questions about probe behavior. In particular, we have used the method to track the assembly, disassembly, and movements of polarity clusters in 2D or 3D. We also present the GUIs that implement this method in a user-friendly manner.

Automated classification of cell motion types

Cells that undergo a series of changes over time, or populations with changing proportions of cells undergoing different morphodynamics, cannot be fully characterized simply by comparing the initial and final states. Visual inspection of movies can be used to assess transient behaviors, but with such manual scoring the transition points between different motion types are not strictly defined, results are subject to personal bias, and throughput is limited. Therefore, we have developed an automated quantification of transient cell dynamics for consistent statistical analysis of many cells. It characterizes cell dynamics at *every* time frame of movies, associating each frame with one of several possible motion types.

Here we combine two measures of cell morphology, the rate of area change and a polarization parameter, which together can be used to characterize major types of cell dynamics. In the example here we focus on differentiating six types of cell morphology changes typically seen in adherent cells undergoing random movement: uniform spreading,

uniform shrinking, polarized spreading, polarized shrinking, polarized movement, and steady shape (non-significant change). The area change is used to classify motion as spreading, shrinking, or neither (constant area); and the polarization parameter is used to classify motion as uniform or polarized. Once the cell outline is determined, the first measure is simply the area difference between two time points. However, the polarization parameter can be defined in a number of different ways:

1. If the cell spreads or shrinks uniformly its centroid does not move. Thus, the velocity of the shape centroid P_1 can serve as a measure for polarized movement. This approach works well for cells with simple shapes and significant centroid displacement from frame to frame.

2. Another way to introduce a polarization parameter, for cells with more complex geometries, or with more subtle dynamics, is to identify boundary points at frame T +dT that protruded (lay outside) or retracted (lay inside) the boundary at time T, where dT is the time lag parameter (Fig. 1A,B). The measure of how uniformly the protruding or retracting boundary points are distributed along the cell perimeter is the measure of cell polarization at a given time. The Kolmogorov-Smirnov test can be used to quantify deviation of the distribution of the protruding (or retracting) boundary points from a uniform distribution. We consider polarization parameters based on protruding boundary points, because our method was developed for studying proteins that are involved in regulating protrusions. However, all following definitions are directly applicable to retracting boundary points as well. Let $n = 1,2,\ldots N$ be a numerical label of boundary points and $v(n)$ be 1 if the point n is protruding and 0 otherwise, so that $M=\sum_{n=1}^{N} v(n)$ is the total number of protruding points. Then, for the measure of uniformness we can use

$$P_2(T) = \max_m \left| \frac{1}{N} \sum_{n=1}^{mN/M} v(n) - \frac{m}{M} \right|, \text{where} \quad m=1,2,\ldots,M.$$

3. Parameter P_2 does not take into account the amount of protrusion, but only the distribution of protruding points along the cell outline. As an alternative approach the polarization parameter can be defined as the length of the polarization vector formed by summation of displacement vectors of protruding boundary points, $\overrightarrow{D}(n) = [D_x(n), D_y(n)]$. However, there are two complications: 1) simple summation of displacement vectors is dependent on cell geometry, and 2) in order to find displacement vectors we need to establish association between boundary points at times T and T+dT (which is a non-trivial problem in general). The first problem can be resolved by mapping the arbitrary cell boundary onto a circle, so that the polarization vector is

$$\overrightarrow{P}_3(T) = \left[\sum_{n=1}^{N} \left|\overrightarrow{D}(n)\right| cos\left(\frac{2\pi n}{N}\right) / \sum_{n=1}^{N} v(n), \sum_{n=1}^{N} \left|\overrightarrow{D}(n)\right| sin\left(\frac{2\pi n}{N}\right) / \sum_{n=1}^{N} v(n) \right].$$

The simplest solution to the second problem is to measure displacement as the minimal distance from the boundary points at time T+dT to the boundary at time T (Fig. 1C,D). Alternatively, the displacement vectors could be determined, for example, by using a physical model with identical springs connecting points between the two boundaries and finding the spring distribution that minimizes the total energy (Allen, Tsygankov, Zawistowski, Elston, & Hahn; Machacek & Danuser, 2006). However, these approaches are still hard to apply to cells with highly dynamic protrusions such as filopodia. In such cases, a pre-processing step that extracts the underlying cell body might be required(Tsygankov et al., 2014).

4. Mapping the cell boundary onto a circle (Fig. 2), the uniformness also can be defined as the length of the vector

$$\overrightarrow{P}_4(T) = \left[\frac{\pi}{N}\sum_{n=1}^{N} v(n)\, cos\left(\frac{2\pi n}{N}\right), \frac{\pi}{N}\sum_{n=1}^{N} v(n)\, sin\left(\frac{2\pi n}{N}\right)\right].$$

This definition is significantly different from the definition in 2 because P_2 reaches its maximum value when there is a single protruding point on the boundary and $\left|\overrightarrow{P}_4\right|$ is largest when half of the boundary points protrude all on one side of the circle.

5. So far we have not considered the directional persistence of the polarization. Indeed, if a polarization vector (such as $\overrightarrow{P}_3$ or $\overrightarrow{P}_4$) changes direction from one side to the opposite and back over several time frames, the cell would not move significantly even though the length of the polarization vector would remain large. Thus, the parameter for persistent polarization can be defined as the length of the vector

$$\overrightarrow{P}_5(T) = \frac{1}{2w+1}\sum_{\tau=-w}^{w}\overrightarrow{P}_3(T+\tau) \quad \text{or} \quad \overrightarrow{P}_6(T) = \frac{1}{2w+1}\sum_{\tau=-w}^{w}\overrightarrow{P}_4(T+\tau),$$

where w is the half-width of the averaging window.

Using the rate of area change R_a and one of the polarization parameters above $\left|\overrightarrow{P}_i\right|$, the cell dynamics can be represented as a trajectory in the parameter space $\left\{R_a, \left|\overrightarrow{P}_i\right|\right\}$. Then, choosing two critical values, R_{cr} and P_{cr}, we split the parameter space into 6 regimes (Fig. 3) corresponding to uniform spreading $\left(R_a \geq R_{cr}, \left|\overrightarrow{P}_i\right| < P_{cr}\right)$, uniform shrinking $\left(R_a \leq -R_{cr}, \left|\overrightarrow{P}_i\right| < P_{cr}\right)$, polarized spreading $\left(R_a \geq R_{cr}, \left|\overrightarrow{P}_i\right| \geq P_{cr}\right)$, polarized shrinking $\left(R_a \leq -R_{cr}, \left|\overrightarrow{P}_i\right| \geq P_{cr}\right)$, polarized movement $\left(-R_{cr} < R_a < R_{cr}, \left|\overrightarrow{P}_i\right| \geq P_{cr}\right)$, and finally the steady shape $\left(-R_{cr} < R_a < R_{cr}, \left|\overrightarrow{P}_i\right| < P_{cr}\right)$. Thus, at any time point the cell dynamics is

classified as one of the motion types according to the location in the parameter space (Supplemental Movie).

Basically the movement of the cell is represented as a *trajectory* in parameter space, where each frame of a movie is represented as a different point in the parameter space shown in Figure 3 and 4.

GUI for morphodynamics classification and ready representation of changes in shape classification over time

To make our method easy to use, we built a Graphical User Interface, *SquigglyMorph*, which requires MATLAB, but does not require direct interaction with the code (Fig. 4). The GUI allows import of data as TIF stacks of cell masks. For the pre-processing step, segmenting cells and acquiring cell masks, any image processing software, such as ImageJ or MetaMorph, will work. However, for the sake of completeness, we included a simple segmentation module *MovThresh*, originally developed as a part of the *CellGeo* package for quantification of cellular protrusion dynamics(Tsygankov et al., 2014). Accurate delineation of the cell boundary is critical; the quality of our analysis is dependent on the quality of the original image and cell segmentation.

After importing a movie, *SquigglyMorph* displays the cell outline and its geometric centroid and allows for visual inspection of the cell dynamics using the time slider. Depending on the data, the user might want to adjust the time lag parameter d*T* and the smoothing window size *W* before the next processing step. The processing (initiated by the “Process” button) includes calculation of the rate of area change R_a and polarization indexes $\left|\overrightarrow{P}_i\right|$.

The processing activates five display windows. The first window displays protruding (white) and retracting (black) areas for every time point *T* with respect to time point *T*+d*T*. The second window shows the distribution of protruding and retracting boundary points along the boundary as being mapped onto a circle. The red dot on the circle in window 2 corresponds to the red dot on the cell boundary in window 1. The third and fourth windows show the rate of area change and polarization index as a function of time (red) and the running average of these parameters (blue) with Gaussian smoothing window of the user-specified size *W*. Finally, the fifth window displays the parameter space (rate of area change vs. polarization index) and the cell dynamics represented as the trajectory through six possible parameter regimes and the corresponding motion types (as stated above the graph). The green dot (and green vertical lines in windows 3 and 4) indicates the current time point, so that users can move the time slider and visually assess the results of automated classification.

At this point users can interactively adjust the values of the critical parameters, R_{cr} and P_{cr}, and switch between different polarization indexes, $\left|\overrightarrow{P}_i\right|$, to examine how these parameters affect results and to provide a better understanding of what factors influence users’ visual assessment of cell dynamics. Users can save the session (the current choice of parameters and the results) for later reloading if needed and also export the result to a text file.

The idea behind such a GUI design is to give users an opportunity to tune the parameters of the classification algorithm (i.e. manually train the algorithm) to a satisfactory level using several examples, and then run the analysis for all cells consistently with a fixed set of parameters. This final step of batch processing is available in Tools->Batch_Processing of the main menu.

As an additional option, the “Analysis” panel allows the user to display the mean square displacement (MSD) of the cell centroid in a new figure. The default units for MSD calculation are pixel and time frame, but this can be changed by specifying the temporal and spatial scales on the right side of the panel.

Importantly, note that the blue line in the lower right hand window represents the changing behavior of the cell over time, showing with a simple graphical representation how the cell transformed from one movement type to another during the course of the experiment. Complex behavior can be represented, stored and analyzed based on this simple representation in 2D space.

Results of morphodynamics classification

Figure 5 illustrates the results of our classification method for a cell that undergoes a series of shape transformations (best seen in the Supplemental Movie). Every 12th frame of the movie is shown in Fig 5A. The dynamic classifications are shown in Fig 5B for two of the polarization measures ($\left|\overrightarrow{P}_5\right|$ and $\left|\overrightarrow{P}_6\right|$). In this figure, every time point is colored according to the type of the cell motion (cyan - uniform spreading, yellow - uniform shrinking, magenta - polarized spreading, green- polarized shrinking, blue - polarized movement, and red – no change in shape). The differences in the two measures are not surprising because the polarity measures $\left|\overrightarrow{P}_1\right|$, $\left|\overrightarrow{P}_3\right|$, and $\left|\overrightarrow{P}_5\right|$ take into account the magnitude of shape change along the boundary, while polarity measures $\left|\overrightarrow{P}_2\right|$, $\left|\overrightarrow{P}_4\right|$, and $\left|\overrightarrow{P}_6\right|$ account only for the distribution of the change along the boundary. Thus, when the cell spreads along the entire boundary (uniformly), but more on one side than the other, the first and the second groups classify the change as polarized and uniform spreading, respectively. Neither of these results is right or wrong, but depends on what information is most important for the study at hand. In the present case, it is our view that the classification given by parameter $\left|\overrightarrow{P}_6\right|$ corresponds most closely to visual inspection of the morphodynamics of this cell. Importantly, for any choice of $\left|\overrightarrow{P}_i\right|$, our method provides a consistent analysis of many cells using precise mathematical definitions.

Geometry-based segmentation of cells in clusters

The human visual processing system is remarkably good at picking out cells even when they are tightly clustered, using a variety of different cues. Computational approaches to solving this “segmentation problem” also take advantage of such cues. For example, by imaging fluorescently labeled nuclei and membrane simultaneously, one can segment cells with an

algorithm that uses the nucleus as a starting indicator of each individual cell, and then fills the space between the nucleus and membrane markers to recover cell shapes. Even if such labeling is not an option, segmentation can succeed by exploiting expectations about cell shape when cells have a simple geometry. For instance, Fig. 6A shows a cluster of cells of the budding yeast *Saccharomyces cerevisiae* expressing a fluorescent probe that concentrates at polarity sites and mother-bud necks. After thresholding, the corresponding binary image in Fig. 6B lacks intensity cues for the location of cell borders, yet the human eye can readily identify the cells by relying purely on geometric cues.

The following geometry-based segmentation method enables automatic segmentation of cells in clusters that can then be tracked over time. The method includes 3 main steps. First, a binary image is obtained by thresholding to eliminate the extracellular background noise (Fig. 6B). Second, a distance map of the binary image is built so that each non-background pixel gets a value equal to the minimal distance from this pixel to the background. The fact that yeast cells are closed ovals (geometric cue) ensures that the distance map has local peaks at the centers of individual cells (Fig. 6C). Third, watershed segmentation is applied to the distance map (Fig. 6D-F). Because the built-in MATLAB function for watershed segmentation (*watershed*) produced unexpected cuts through some of the cells (Fig. 6D), we used our own implementation of the watershed algorithm. This over-segmentation problem is well known in watershed applications and typically addressed by post-processing merging of segmented regions(Adiga & Chaudhuri, 2001; Long, Peng, & Myers, 2007; Najman & Schmitt, 1996). In contrast, in our algorithm regions incrementally grow starting from the local intensity maxima, so that when two regions meet, the algorithm either indicates the pixel between the regions as the border or merges the smaller region with the larger one, depending on whether the size of the smaller region S is larger or smaller than a given parameter, S_{cr}. For $S_{cr} = 0$, the result has the same artifacts as the MATLAB's *watershed* (Fig. 6E), however, for $S_{cr} = 1$ pixel, all the unwanted artifacts disappear and segmentation works as expected (Fig. 6F). In principle, the described approach does *not* require that the cells have circular or elliptical shapes as long as there are enough empty spaces (holes) in places where three or more cells come together. However, it is unlikely that cells that can assume an arbitrary shape (e.g. some mammalian cells) would systematically leave such spaces in their clusters. Thus, the method is most obviously applicable to clusters of units with simple definite shapes such as yeast cells, bacterial cells, red blood cells, or the nuclei of mammalian cells.

Because the results of the segmentation routine depend on the initial choice of the threshold value, and we do not have a way to assess the quality of segmentation other than by visual inspection (Fig 6A and G), our approach is to start by running segmentation for a number of threshold values within a manually specified range and then choose a threshold value H_{ref} for one of the timepoints T_{ref} as a reference for automatic tracking through the other frames. Let $L(T,H)$ be a numerical label of cells at time T as segmented with the threshold H. The tracking objective is to find a sequence of labels $\{L(1,H_1), L(2,H_2),\ldots, , L(N,H_N)\}$ for each $L(T_{\text{ref}},H_{\text{ref}})$. In this way, two different cells at frame $T_{\text{ref}}+1$ that correspond to two different cells at $(T_{\text{ref}},H_{\text{ref}})$, might come from segmentation at two different threshold levels. In order to establish such correspondence, for a given cell n at (T,H) we minimize the mismatch

$$E_{n,m}^{H,H'}(T) = \sum_{x,y} \left| I_{n(T,H)} - I_{m\left(T+1,H'\right)} \right|$$

over all cells m and all threshold values H' at time T+1, where $I_{n(T,H)}$ is the binary mask of the cell n at (T,H). The minimization procedure is repeated for every cell at the reference frame (T_{ref},H_{ref}) iteratively starting from the reference frame forward and backward in time.

Once segmentation and tracking are completed, we overlay cell masks with the original images (Fig 6G) (by tracking we mean identifying the same cell is it occurs in different frames of a movie, even when its size and position have changed over time). We can then employ internal thresholding to identify all protein clusters (connected objects) within the cells (Fig. 6H). Internal thresholds can be set manually or can be determined automatically as, for example, mean intensity plus 2 standard deviations from the mean. We keep track of the biggest and the second biggest spot to capture the situations when large protein clusters break up into smaller pieces. A variety of quantitative measures can then be used to describe the dynamics of protein localization. For the example of clusters of polarity factors, this can include tracking the size and integrated intensity of the cluster, the velocity or mean square displacement of moving clusters(Dyer et al., 2013), and the coefficient of variation of total cell pixel intensity (indicative of the degree of clustering).

GUI for cell segmentation and quantification of protein clusters

We implemented the above method in a two-part interface, *SegmentMe*, for cell segmentation and quantification of protein clusters dynamics in 2D projection and in 3D space for z-stack data. These GUIs are coded in MATLAB (with supplementary functions in Java) and require MATLAB installation. However, they are designed as an intuitive click-and-drag software interface.

GUI module for 2D analysis

The controls in this module are separated in two blocks: a “Processing” panel and an “Analysis” panel (Fig. 7A). All the controls are deactivated until users import a movie as a TIF stack. If the original data exists as a collection of z-stack files for each time frame, users can use one of the available image processing tools, such as ImageJ, to create average or maximum projections from the z-stacks and then combine individual time frames into a single TIF stack.

Upon importing the movie, the processing controls become activated successively, starting with the noise reduction step that includes two types of smoothing: three-frame time averaging (“Dynamic Filter” button) and spatial averaging (“Gaussian Filter” button). For the Gaussian filter, users can specify the filter size and standard deviation. This step can be skipped if the signal-to-noise ratio is high enough. The next processing step (activated by the “Threshold” button) allows users to set the range and number of threshold values. The idea here is to choose a range, which is broad enough to ensure that there are optimal threshold values within the range for each part of the image and each frame of the movie. The tracking algorithm will scan through the different threshold values to find the best correspondence

with each cell in the reference frame, as described in the previous section. Choosing a range that is too broad and/or a number of threshold values that is too large is counterproductive, because it will make processing unnecessarily slow without any gain in the quality of segmentation. A good strategy is to use both the vertical threshold slider and the horizontal time slider to explore the thresholding effects at different time points before setting the range and the number of steps. The next step is to run segmentation ("Segment" button) and explore the results (as displayed in the bottom right window, Fig. 7A) using the vertical threshold slider and the horizontal time slider to choose the reference frame ("Set Reference" button), i.e. the time point and threshold value for which most of the cells are segmented accurately enough. After the reference image is set, the tracking can be initiated by clicking the "Track" button.

Upon completion of the tracking, the "Analysis" panel becomes activated. The top left window now displays the original data before noise reduction (Fig. 7B). At this point the results of segmentation and tracking can be saved either to reload later when or to be imported by the other module for 3D analysis. Individual cells can be accessed in two ways: a horizontal cell slider highlights cells in the order from the largest to the smallest; alternatively, activating "Cell Select" from the tool bar below the menu allows the user to drag the red target marker over any cell of interest, and de-activating "Cell Select" completes the selection. The highlighted cells are also displayed individually in the bottom right corner for closer inspection (Fig. 7C).

To analyze the dynamics of protein clusters (intensity spots) in the selected cells, users need to switch to the "Threshold" view in the "View" panel and either activate "auto" thresholding or select a fixed threshold value using the vertical slider (Fig. 7A). The automatic thresholds are calculated as the mean plus two standard deviations of the intensity values inside the cell. The size filter checkbox at the bottom of the "Analysis" panel filters out all spots smaller than the user-specified size in square pixels. Any of three different measures: the mean square displacement of the centroid, velocity of the centroid, and spot size, can be displayed in a new figure by clicking the corresponding buttons. The GUI will display these measures either for the **f**irst (largest) spot "F", the **s**econd to largest spot "S", or **a**ll the spots together "A" depending on the users' choice of corresponding radio buttons. In the bottom right window for the individual cell view, the centroids of "F", "S" and "A" spots are marked by the red, blue and black dots, respectively (Fig. 7C). The intensity coefficient of variation can be also displayed in a new figure ("CV" button). All measurements for the selected cell can be saved as a text file by clicking "Save Cell Data". It is important to specify the spatial and temporal scales before saving; otherwise the measurements will be saved in pixel and frame units.

GUI module for 3D analysis

This module requires two imports. The first is a MATLAB file (.mat) generated by the module for 2D analysis. This file contains the results of segmentation based on the 2D projection. For cells clustered on a substrate in a single-cell layer, it is sufficient to make a simple cylindrical extension of the 2D segments (i.e. apply the same masks to each z-slice) to segment cell data in 3D space. The second file that needs to be imported is a TIF file with

the z-stack data for the first time frame. The other TIF files with the z-stack data for all other time frames will be imported automatically as long as they are in the same folder and named/numbered consistently.

The projected segments are displayed in the top left window (Fig. 7D) and cell selection can be done with the cell slider or the “Cell Select” tool in the same manner as in the module for 2D analysis. The 3D data for the selected cell can be displayed in the top right window by clicking the “Load 3D” button. Two sliders below the window then become active and allow the user to monitor each z-slice and time point (Fig. 7E). The bottom right window is for three-dimensional visualization of protein clusters (intensity spots) inside the cell. The GUI generates isosurfaces for all 3D spots with intensity higher than a user-defined threshold value (Fig. 7F). The threshold can be specified either automatically (based on the mean plus two standard deviations intensity) using the “auto” checkbox or manually using the vertical threshold slider. Small spots can be taken out of consideration using the “Small Spot Filter” panel, where critical spot size is specified in cubic pixels.

All other controls in the GUI are analogous to the GUI for 2D analysis (Fig. 7A) and designed for visualization and saving various 3D measurements for dynamics of the first (largest) spot, the second to largest spot, or all the spots together. To switch to another cell, the user needs to click the “Select Again” button, make a new selection, and click “Load 3D” again (Fig. 7E).

Results for quantifying protein clusters

Figure 8 shows protein clusters identified in a cell that was segmented using our graphical user interface *SegmentMe*. In this figure every 4^{th} frame is shown. The top panels show the original fluorescence signal and the bottom panels indicate the cell boundary and the detected protein clusters using an internal threshold of the mean + 2 standard deviations (a 10 square pixel filter is applied to eliminate small spots due to signal noise). In this example, the clusters are highly dynamic and undergo frequent splitting and merging events. The measurements successfully captured this behavior (Fig. 8B) as seen by the significant variations in the size of the largest (red lines) and the second largest (blue lines) clusters. Indeed, the ratio of the second largest cluster to the total size of all clusters fluctuates from 0 to ~0.5 throughout the movie (Fig. 8C). The mean square displacement of the largest cluster (Fig. 8D) quickly reaches a plateau, which is characteristic for random motion within a confined area (the cell). Consistently with this behavior, the velocity of the largest spot (Fig. 8E) reaches values over half the cell size per frame (> 25 pixel/frame).

Discussion

Here we presented quantitative methods for analysis of two very different types of cell biology problems involving cell morphology: dynamics of shape change for single cells with complex geometries, and dynamics of protein clusters in tightly packed cells, taking advantage of their simple geometries to obviate the need for additional markers for segmentation.

The major issue in the first system was how to define measures of shape change that can be applied consistently across very diverse cell geometries. In general, there are a large number of geometric parameters that can be used to classify cell shapes. However, our focus here was on cell shape dynamics rather than on static shape. Therefore, for classification of six major types of shape changes covering the typical morphodynamics of isolated motile cells, it was sufficient to use just two parameters: the rate of cell area change and polarization index, using different definitions of the polarization index as described above. We developed a GUI that gives users an opportunity to try different polarization indexes, visualize, and assess the results with the click of a button. Interactive adjustment of critical parameters (border lines between motion types in parameter space) also provides the user with a way to examine the effects of parameter choice on analysis outcome. Importantly, once the parameter choices are made, the classification can be applied consistently to all the cells in the data set through the batch processing capabilities of the GUI.

In the second biological system, the major issue was how to correctly identify individual cells that are in contact with each other in an automated manner. Without additional segmentation probes, the solution is possible, but must rely on geometric cues. Our method consists of a series of processing steps, including a modified watershed segmentation of the distance map. The modification that involves small region merging during watershed segmentation turned out to be critical because the traditional watershed method (for example as implemented in the MATLAB function *watershed*) produced unwanted cuts through some of the cells. Once again, we made the described method available as a GUI, which can be used for cell segmentation and also for quantification of protein cluster dynamics in both 2D and 3D.

Although the GUIs include a variety of quantitative measures, such as mean squared displacement or cluster size, this set might not cover all possible needs of a user. However, the output of the GUIs contains all the necessary information for further analysis that users can tailor to the specific needs of their project.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgements

We thank Michael Guarino and Ziyou Wu for their assistance in optimizing the code for faster performance. We are grateful for funding from NIH (KH: NIGMS P01-GM103723-01; TE: NIH GM079271, NCI 200079604), and the Army Research Office (TE) for supporting this work.

References

Adiga PSU, Chaudhuri BB. An efficient method based on watershed and rule-based merging for segmentation of 3-D histo-pathological images. Pattern Recognition. 2001; 34(7):1449–1458.

Allen RJ, Tsygankov D, Zawistowski JS, Elston TC, Hahn KM. Automated line scan analysis to quantify biosensor activity at the cell edge. Methods. 2014; 66(2014):162–167. [PubMed: 23994242]

Carlton PM, Boulanger J, Kervrann C, Sibarita JB, Salamero J, Gordon-Messer S, et al. Fast live simultaneous multiwavelength four-dimensional optical microscopy. Proceedings of the National

Academy of Sciences of the United States of America. 2010; 107(37):16016–16022. [PubMed: 20705899]

Danuser G. Computer Vision in Cell Biology. Cell. 2011; 147(5):973–978. [PubMed: 22118455]

Doncic A, Eser U, Atay O, Skotheim JM. An Algorithm to Automate Yeast Segmentation and Tracking. Plos One. 2013; 8(3)

Dyer JM, Savage NS, Jin M, Zyla TR, Elston TC, Lew DJ. Tracking Shallow Chemical Gradients by Actin-Driven Wandering of the Polarization Site. Current Biology. 2013; 23(1):32–41. [PubMed: 23200992]

Indhumathi C, Cai YY, Guan YQ, Opas M. An automatic segmentation algorithm for 3D cell cluster splitting using volumetric confocal images. Journal of Microscopy. 2011; 243(1):60–76. [PubMed: 21288236]

Loncaric S. A survey of shape analysis techniques. Pattern Recognition. 1998; 31(8):983–1001.

Long, FH.; Peng, HC.; Myers, E. Automatic segmentation of nuclei in 3D microscopy images of C.elegans; 2007 4th Ieee International Symposium on Biomedical Imaging : Macro to Nano; 2007; p. 536-539.

Machacek M, Danuser G. Morphodynamic profiling of protrusion phenotypes. Biophys J. 2006; 90(4): 1439–1452. [PubMed: 16326902]

Machacek M, Hodgson L, Welch C, Elliott H, Pertz O, Nalbant P, et al. Coordination of Rho GTPase activities during cell protrusion. Nature. 2009; 461(7260):99–103. [PubMed: 19693013]

Malpica N, de Solorzano CO, Vaquero JJ, Santos A, Vallcorba I, Garcia-Sagredo JM, et al. Applying watershed algorithms to the segmentation of clustered nuclei. Cytometry. 1997; 28(4):289–297. [PubMed: 9266748]

Najman L, Schmitt M. Geodesic saliency of watershed contours and hierarchical segmentation. Ieee Transactions on Pattern Analysis and Machine Intelligence. 1996; 18(12):1163–1173.

Nilsson B, Heyden A. Segmentation of complex cell clusters in microscopic images: Application to bone marrow samples. Cytometry Part A. 2005; 66A(1):24–31.

Sailem H, Bousgouni V, Cooper S, Bakal C. Cross-talk between Rho and Rac GTPases drives deterministic exploration of cellular shape space and morphological heterogeneity. Open Biol. 2014; 4(1):130132. [PubMed: 24451547]

Schmitt O, Reetz S. On the Decomposition of Cell Clusters. Journal of Mathematical Imaging and Vision. 2009; 33(1):85–103.

Tsygankov D, Bilancia CG, Vitriol EA, Hahn KM, Peifer M, Elston TC. CellGeo: A computational platform for the analysis of shape changes in cells with complex geometries. Journal of Cell Biology. 2014; 204(3):443–460. [PubMed: 24493591]

Zhang DS, Lu GJ. Review of shape representation and description techniques. Pattern Recognition. 2004; 37(1):1–19.

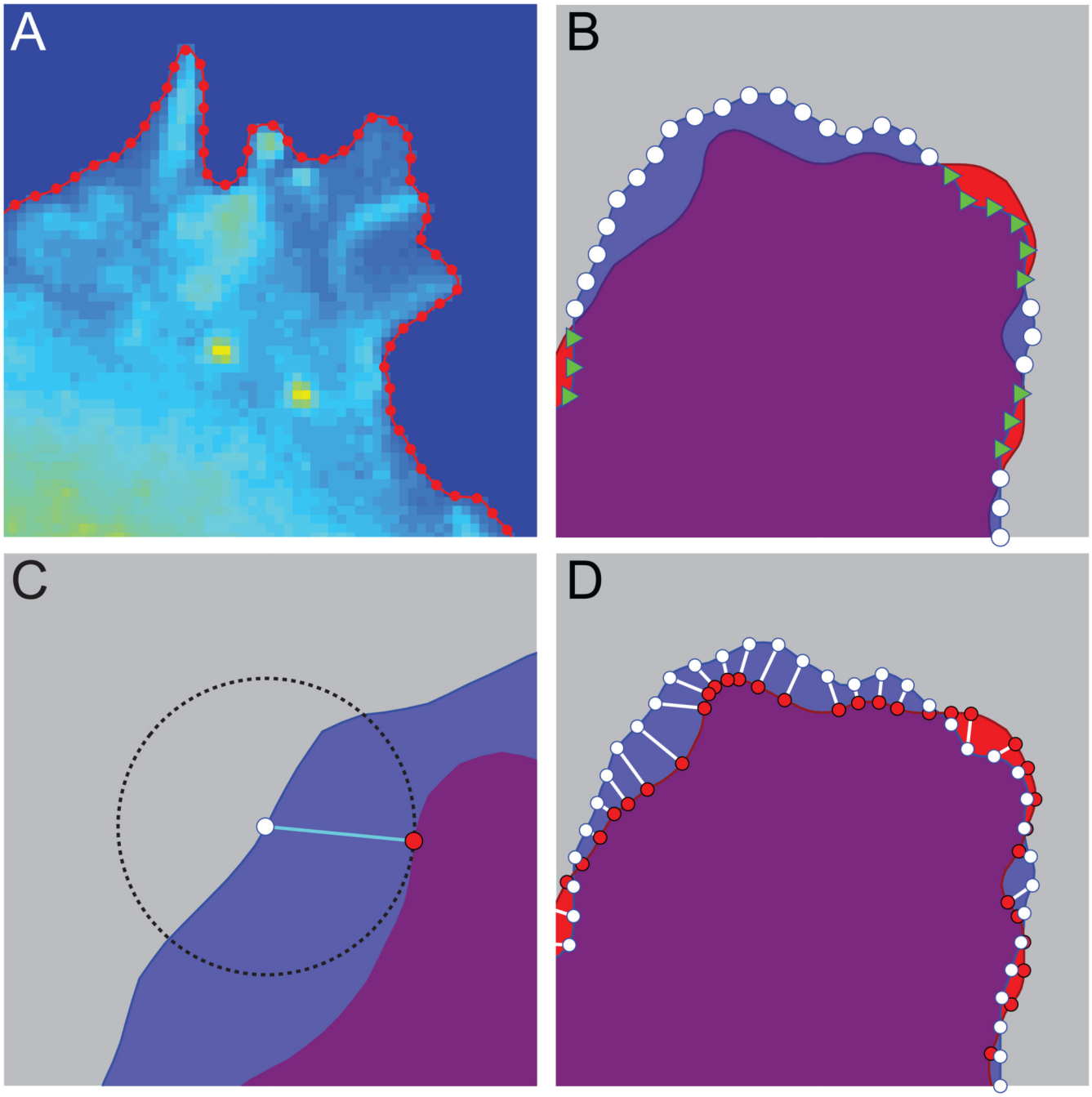

Fig. 1. Boundary characterization

A. Smooth cell boundary (red) obtained by 1) applying a Gaussian filter to the cell binary mask, 2) building a contour (isoline) at the 0.5 level, and 3) equally distributing a specified number of points along the contour. **B.** Two overlaid boundaries from time *T* (red) and *T*+d*T* (blue). Protruding and retracting boundary points at *T*+d*T* with respect to the boundary at *T* are marked as white circles and green triangles, respectively. **C.** The amount of protrusion is measured as a minimal distance (illustrated by dashed circle) from a boundary point at *T*+d*T*

to the boundary at *T*. **D.** Displacement vectors for boundaries in **B** defined as illustrated in **C.**

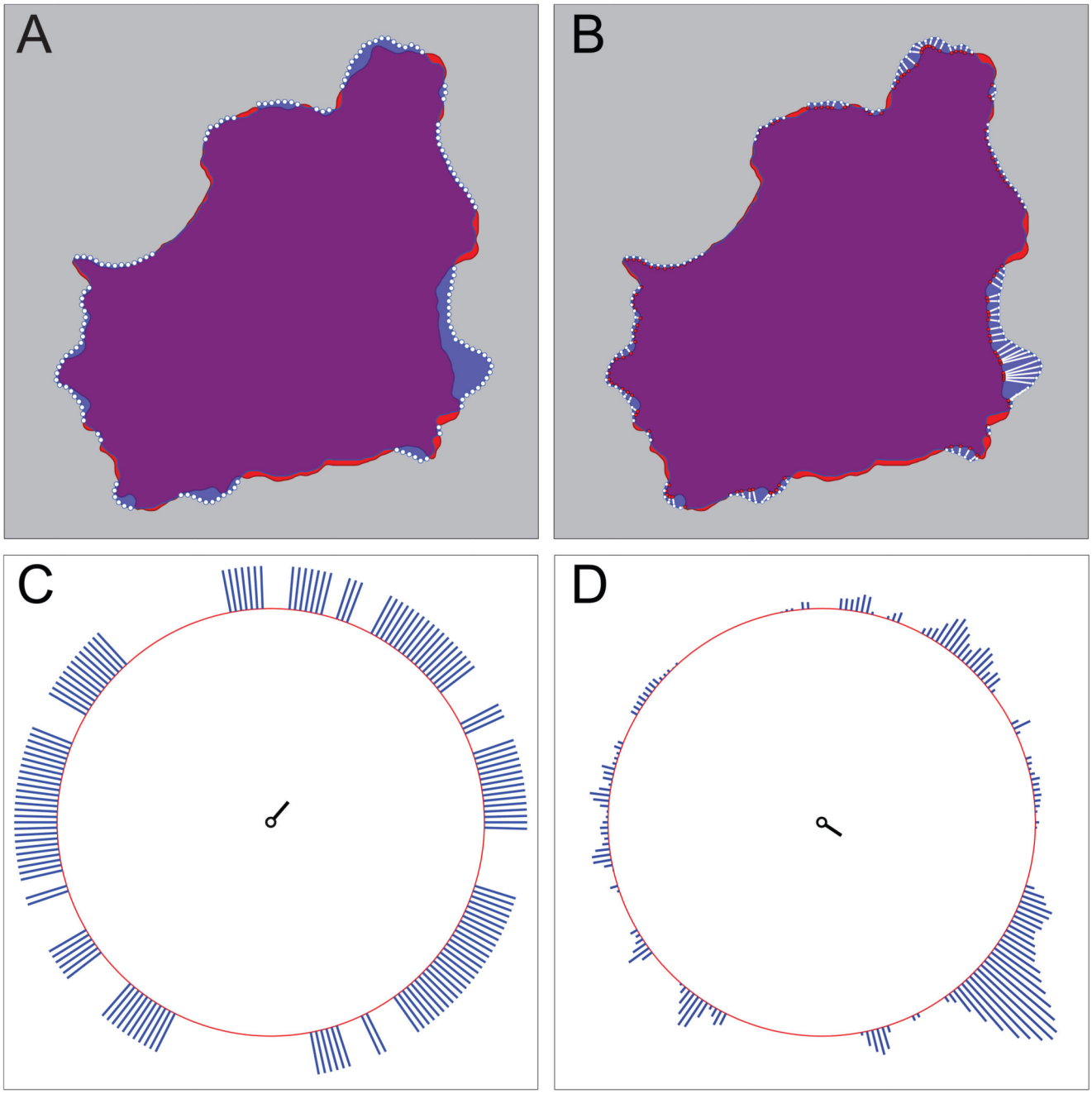

Fig. 2. Illustration of polarization parameters

A,B. Two overlaid boundaries from time frame *T* (red) and *T*+d*T* (blue). **A.** Protruding boundary points are indicated by white circles. **B.** Displacement vectors for protruding boundary points are shown as white lines. **C,D.** Distribution of protruding boundary points from **A,B** mapped on a circle. **C.** Each protruding boundary point contributes a radially directed unit vector (blue lines). Polarization parameter P_4 is the length of the total vector (black) calculated as the sum of all radial vectors. Factor π/N scales the polarization vectors so that its maximum possible length is 1. **D.** Each protruding boundary point contributes a radially directed vector with length equal to the length of the corresponding displacement

vector (blue lines). Polarization parameter P_3 is the length of the total vector (black) calculated as the sum of all radial vectors and normalized by the number of protruding points.

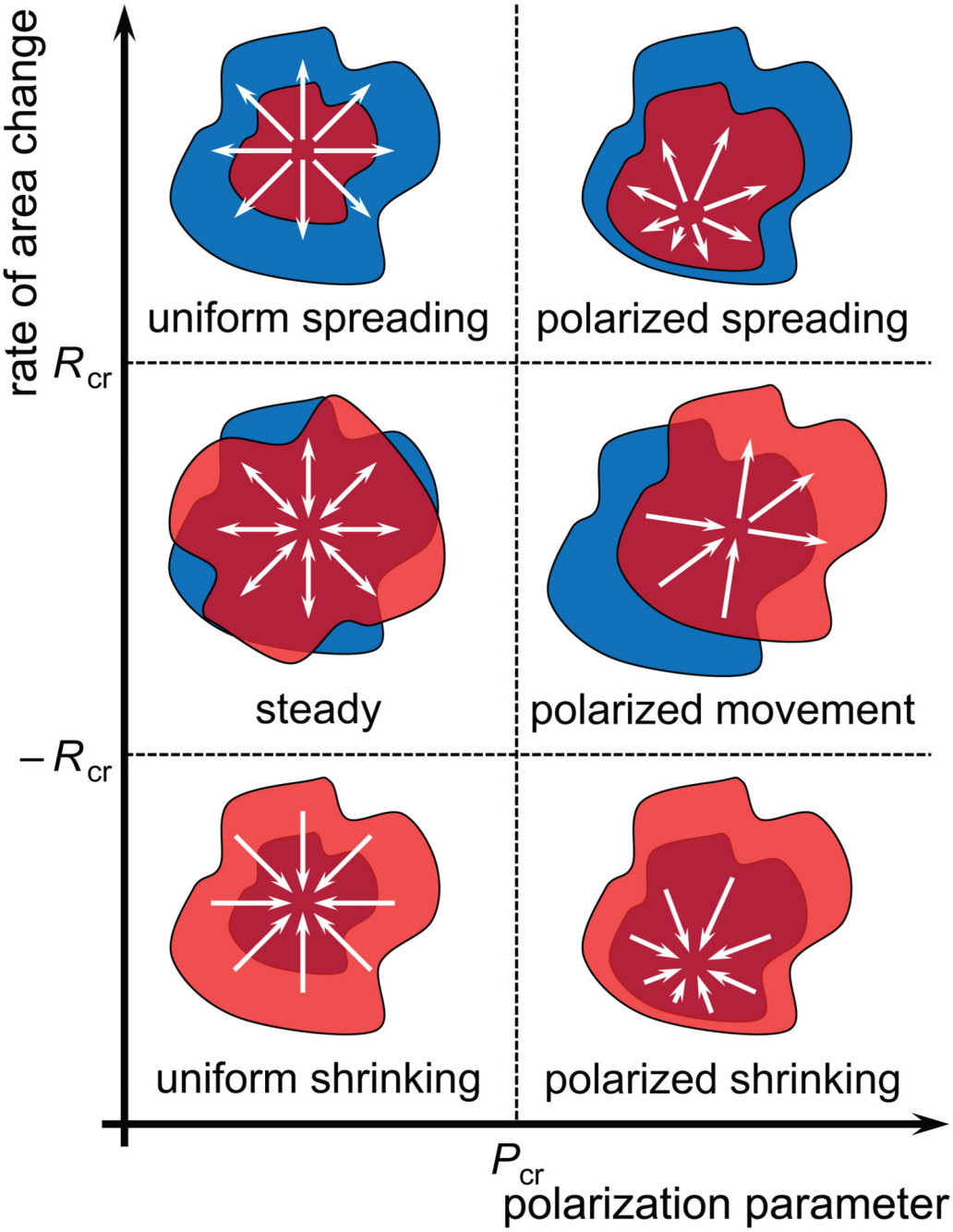

Fig. 3. Illustration of different types of cell shape changes represented by different areas in the parameter space

critical polarization parameter P_{cr} separates uniform and polarized shape changes; critical rate area change R_{cr} separates spreading, shrinking, and non-significant area changes. Thus, a two-parameter classification distinguishes uniform spreading, polarized spreading, uniform shrinking, polarized shrinking, polarized movement, and steady shape.

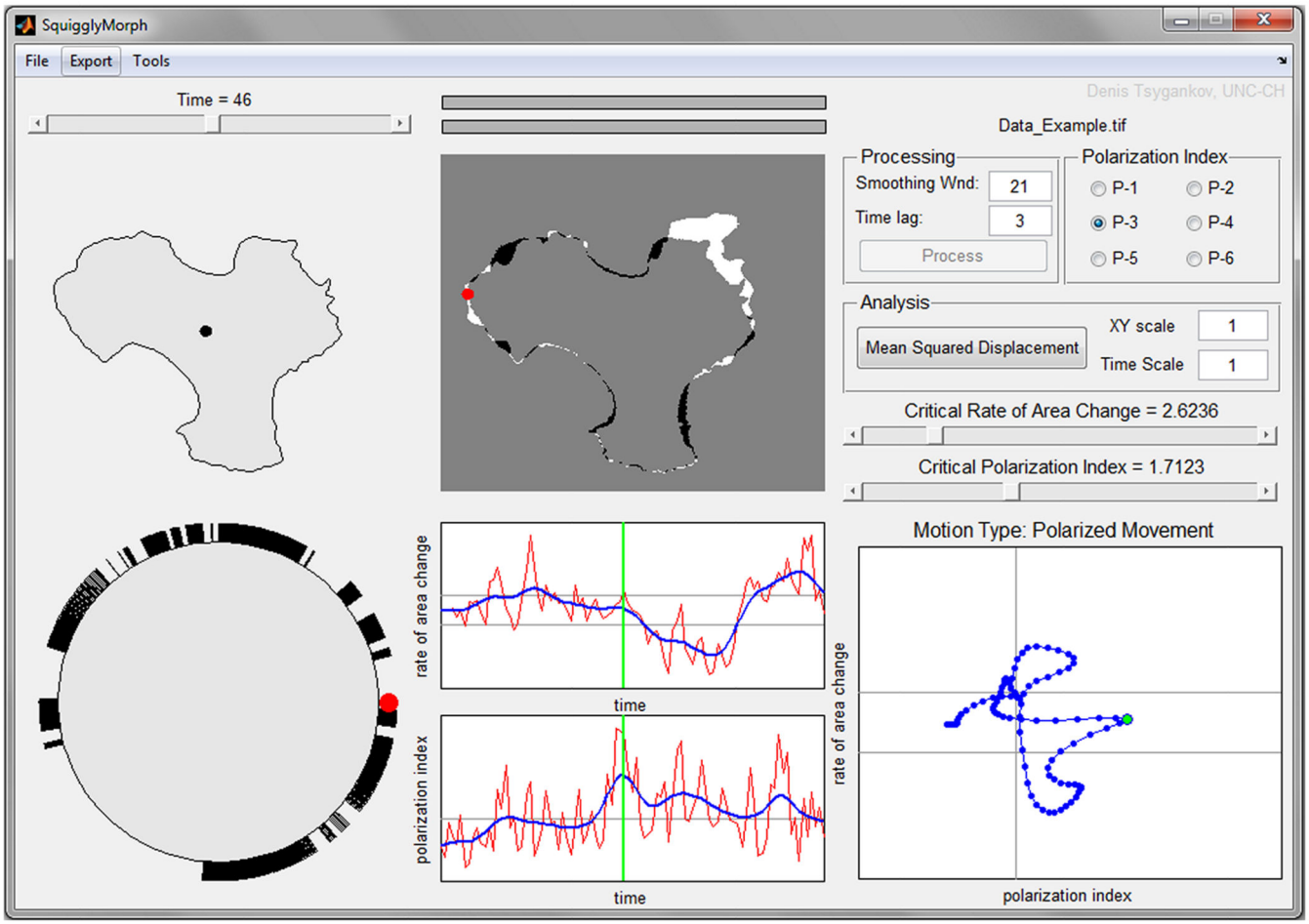

Fig. 4. Screenshot of the graphical user interface, *SquigglyMorph*, for classification of cell motion types

The GUI displays cell boundary, protruding and retracting cell parts, distribution profile of protruding boundary points, rate of area change and polarization parameter as functions of time, and the morphodynamics trajectory in parameter space. The GUI also includes controls that adjust critical parameters for tuning the classification procedure.

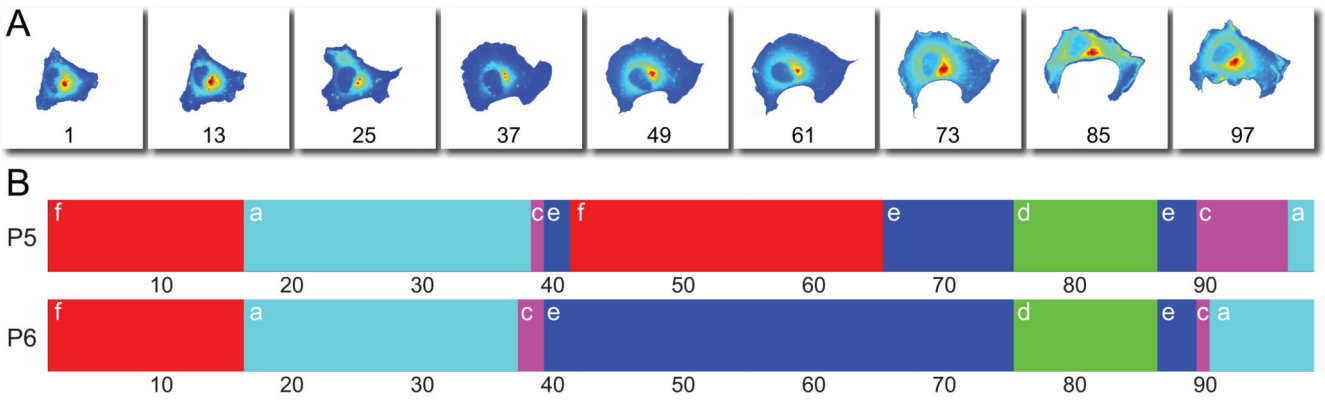

Fig. 5. Analysis of cell motion types

A. 9 time frames from a cell movie (Supplemental Materials). **B.** Motion types as a function of time using two different polarizations parameters ($\left|\vec{P}_5\right|$ and $\left|\vec{P}_6\right|$). Motion types are colored according to the following scheme: cyan (uniform spreading), yellow (uniform shrinking), magenta (polarized spreading), green (polarized shrinking), blue (polarized movement), and red (steady/non-significant change). The differences in cell movement indicated by the classification are best seen in the movie.

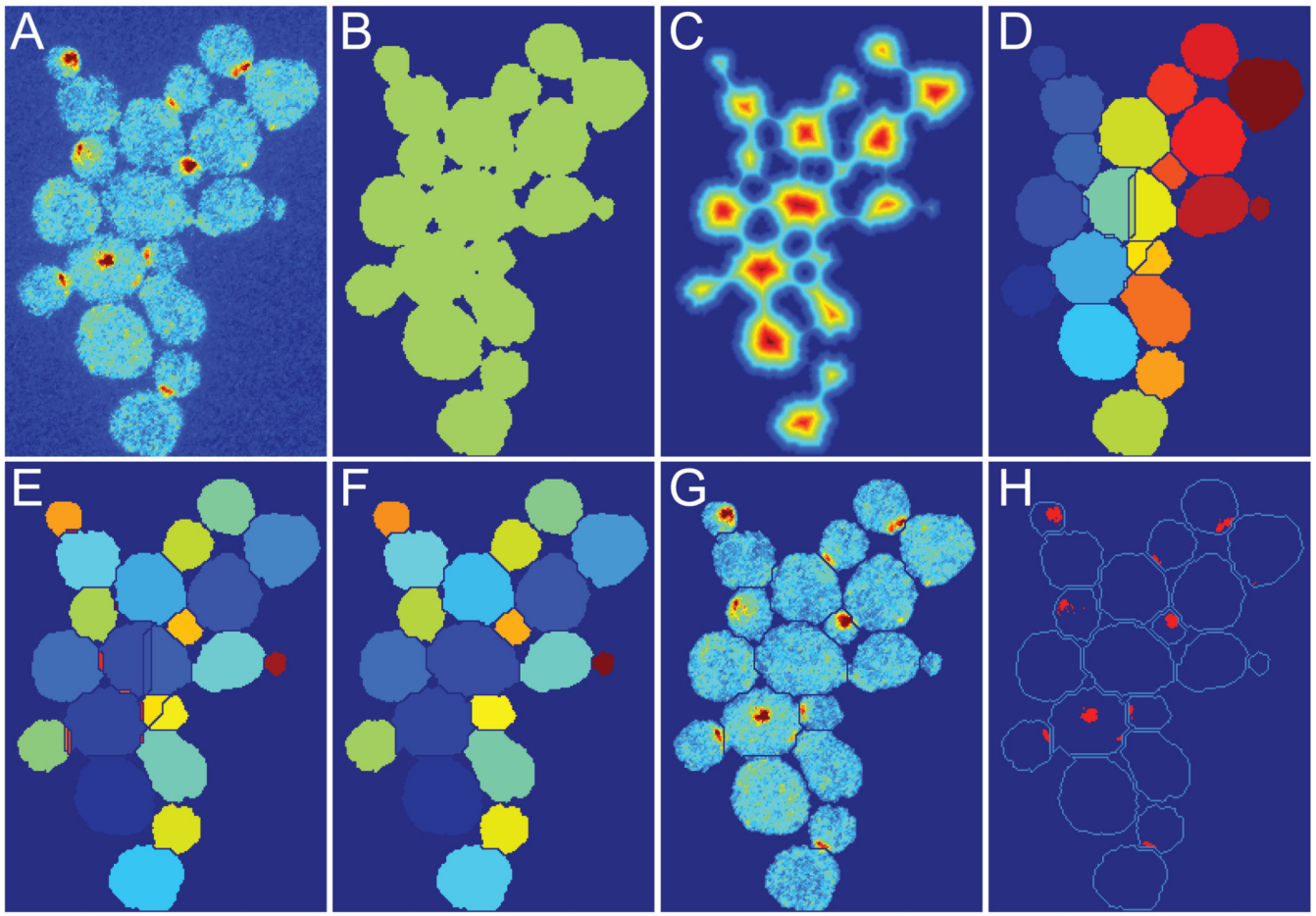

Fig. 6. Segmentation of cells in clusters

A. Original (unprocessed) image. **B.** Thresholded (binary) image. **C.** Distance map of the image in **B. D.-F.** Watershed segmentation of the distance map in **C** using the built-in MATLAB function (**D**), our algorithm without merging (**E**), and our algorithm with merging (**F**). Only the algorithm with merging (**F**) provides proper (visually expected) segmentation. **G.** Original image overlaid with segmented cell masks. **H.** Segmented cell outlines (light blue) and internally thresholded protein clusters (red).

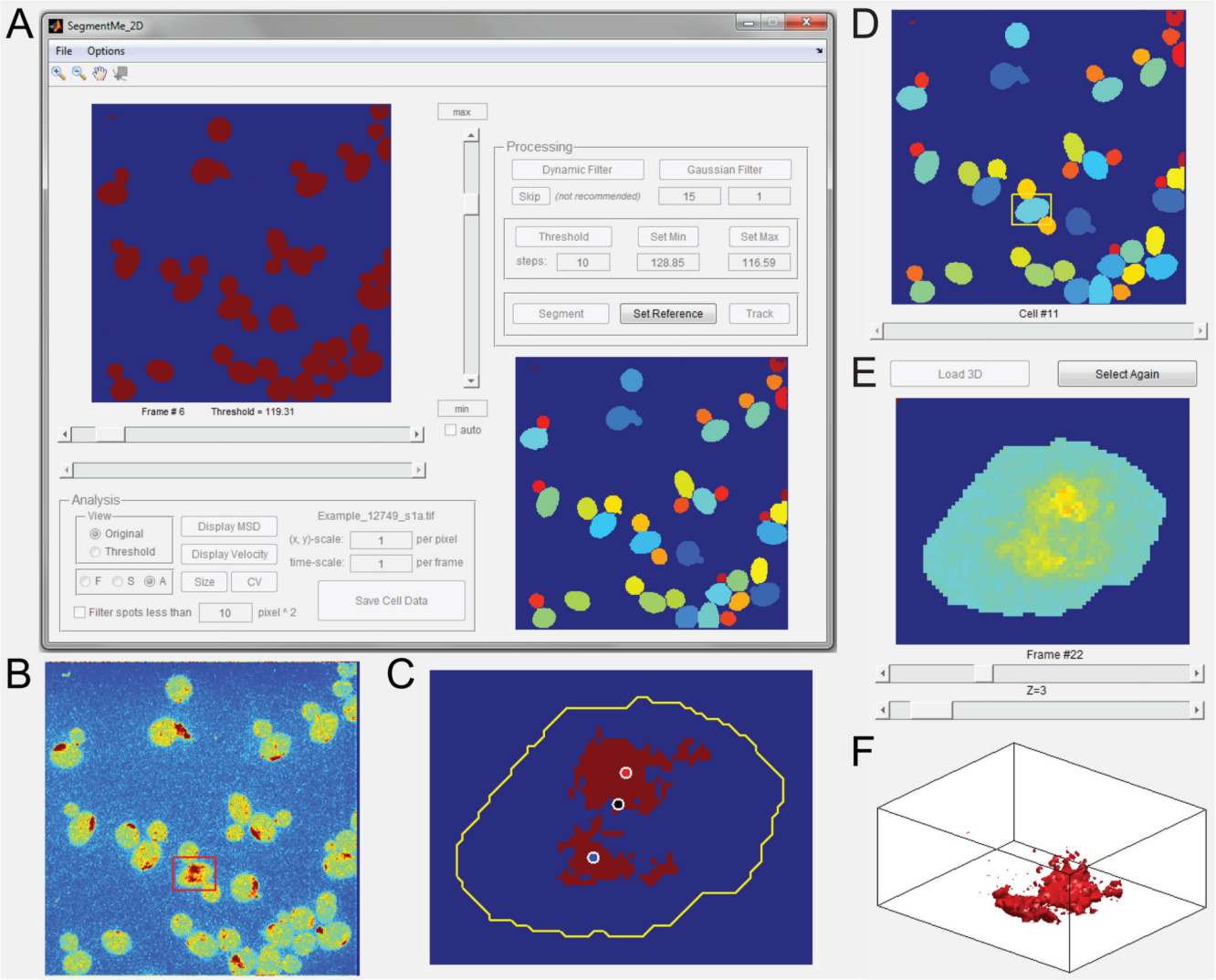

Fig. 7. Screenshot of the graphical user interface, *SegmentMe*, for protein cluster quantification in 2D and 3D

A. Screenshot of the processing step in the graphical user interface, *SegmentMe*, for cell segmentation: Bottom right panel shows the result of segmentation of the binary image at the top left panel. **B,C.** Screenshots from the analysis step. **B.** An interactively selected cell in the original image highlighted by a red box. **C.** The selected cell after segmentation, tracking and internal thresholding. Red, blue, and black circles indicate centroids of the largest, second to largest, and all clusters, respectively. **D-F.** Screenshots from the protein cluster quantification step in 3D. **D,E.** An interactively selected cell (highlighted with a yellow box in **D**). **E.** The transition between time frames and z-stack levels is controlled by two sliders under the cell image. **F.** The panel that shows 3D spots (as isosurfaces) defined by internal thresholding (with automatically or manually chosen threshold value).

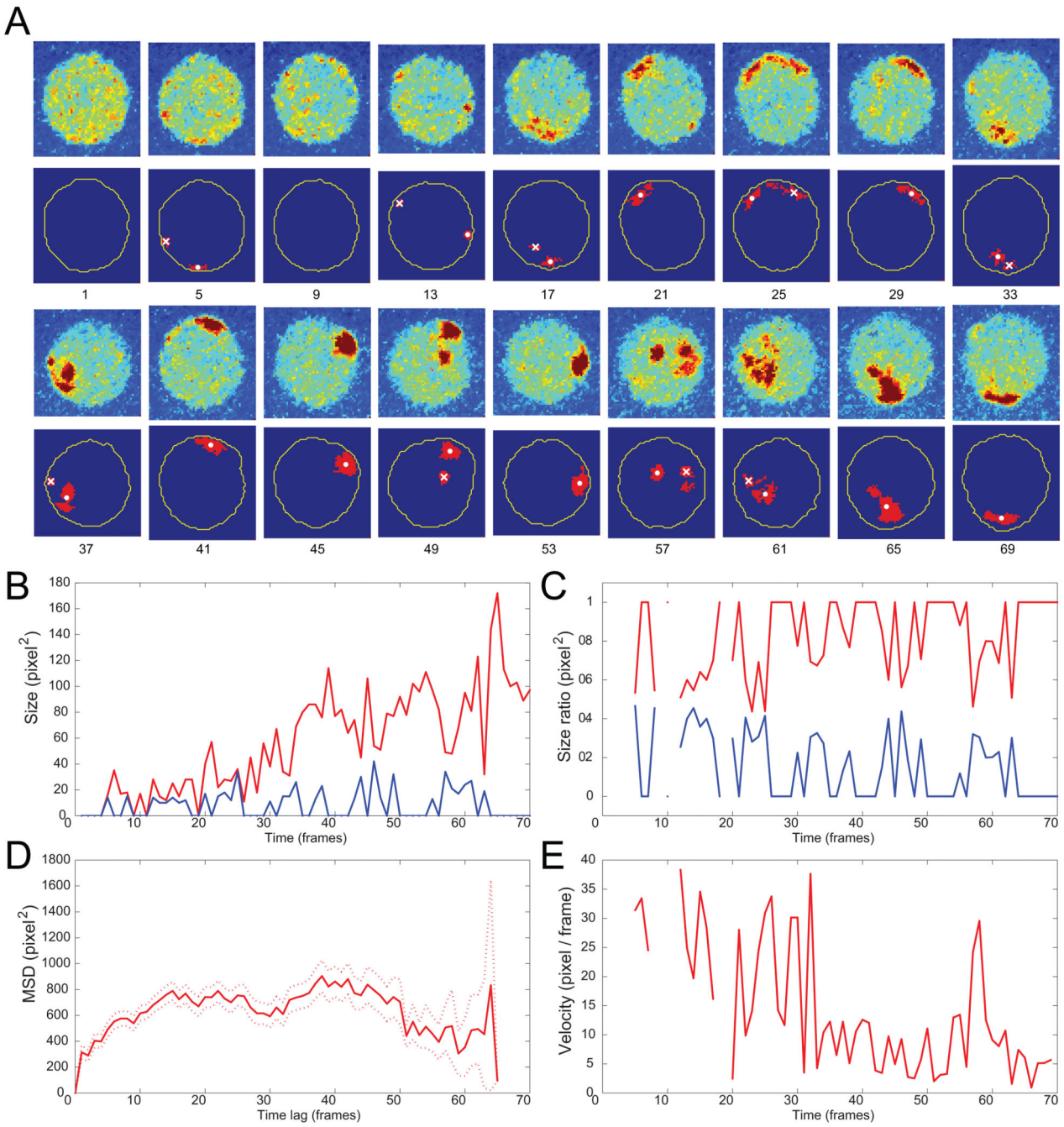

Fig. 8. Analysis of protein cluster dynamics using *SegmentMe*

A. 18 time frames of one of the cells in a movie after cell segmentation and tracking. Top panels show the original image cropped around the cell. The bottom panes show only the boundary of the cell (yellow) and bright spots (protein clusters). The threshold for cluster identification is found as mean intensity of all cell pixels plus two standard deviations. White dots and crosses indicate the centroids of the largest and the second largest spots, respectively. **B.** The size of the largest (red) and the second largest (blue) spots as a function of time. **C.** The ratio of the largest (red) and the second largest (blue) sports to the total size

of all spots as a function of time. **D.** The mean squared displacement of the largest spot as function of time lag. Doted lines indicate plus/minus standard error. **E.** The velocity of the largest spot as a function of time.