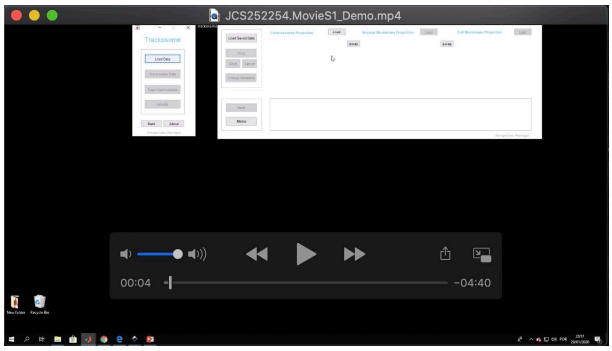
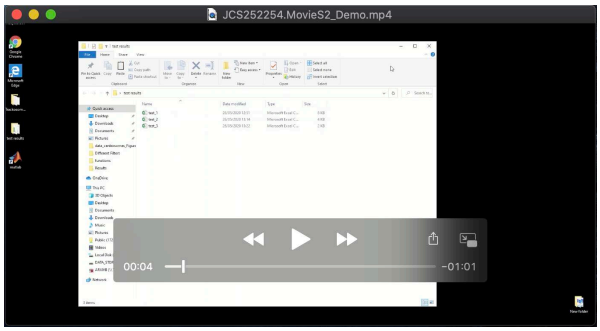


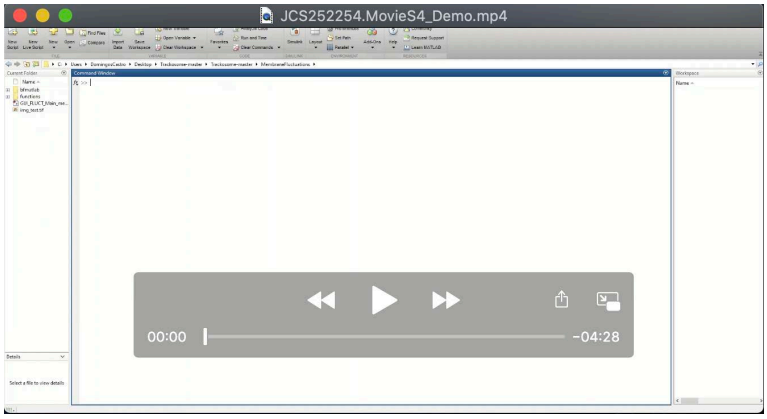
Movie 1. Demonstration of Centrosome Dynamics module of Trackosome. Demonstration of how to: load 3D live-imaging data from centrosomes, nuclear membrane and cellular membrane; perform centrosome tracking and correct tracking mistakes; explore and save the results; reload previously saved data.



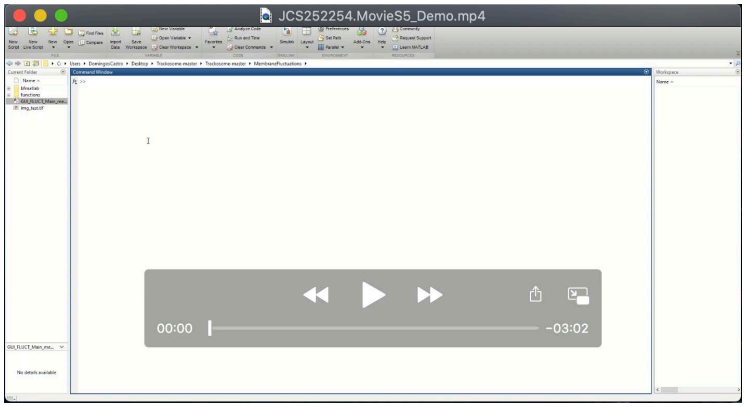
Movie 2. Demonstration of results compilation in Centrosome Dynamics module of Trackosome. Demonstration of how to compile the results from different files analyzed with the Centrosome Dynamics module into a single excel file.

Movie 3. Demonstration of the membrane corrections in Centrosome Dynamics module. Demonstration of how to correct errors in the segmentation of the nuclear membrane and cellular membrane in Centrosome Dynamics module.

Journal of Cell Science • Supplementary information

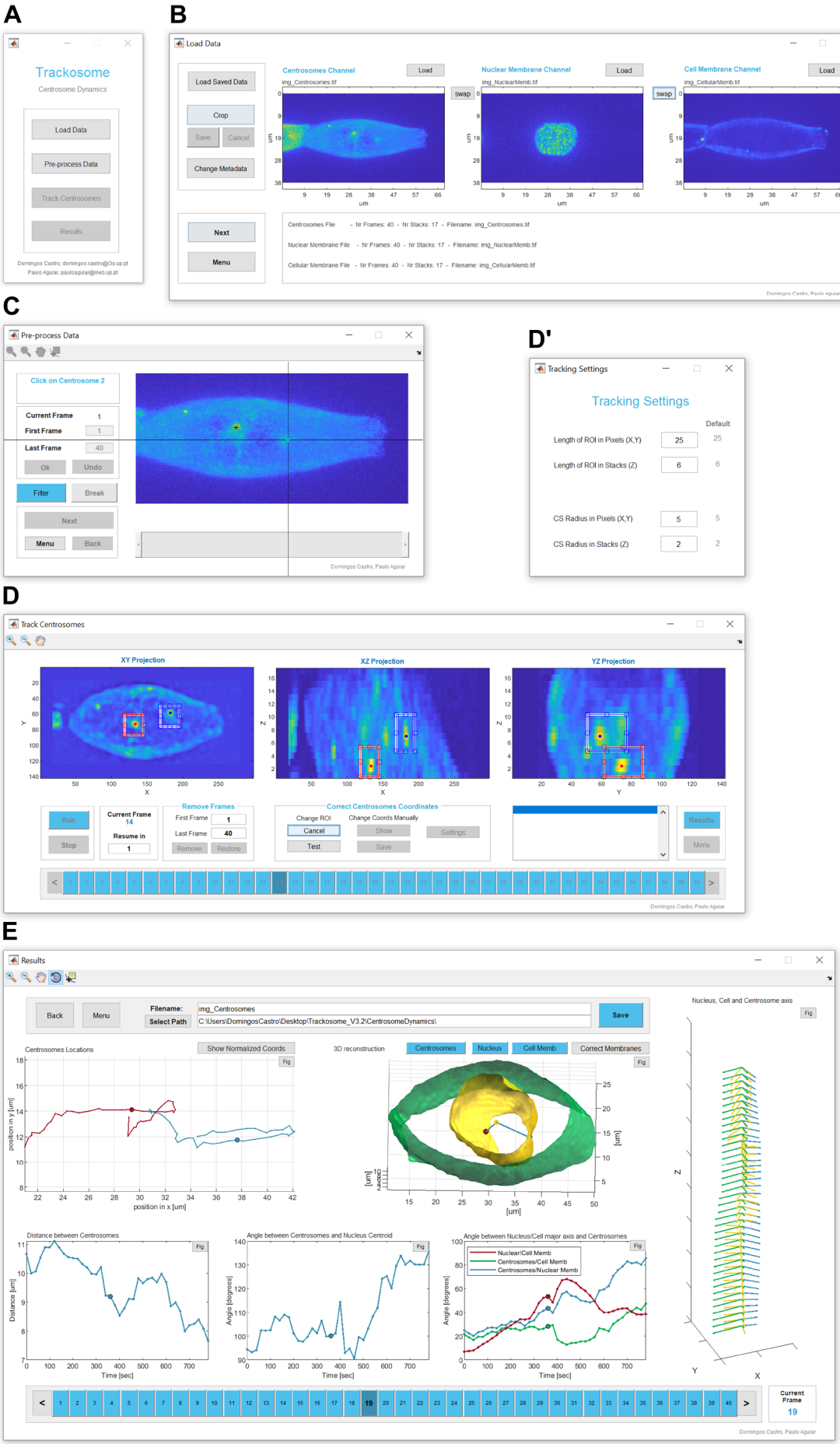


Movie 4. Demonstration of Nuclear Envelope Fluctuations module of Trackosome. Demonstration of how to: load files from 2D live-imaging data of the nuclear envelope; segment the nuclear envelope and correct segmentation mistakes; analyze, correct and save fluctuation results; reload previously saved data.



Movie 5. Demonstration of batch mode analysis in Nuclear Envelope Fluctuations module of Trackosome. Demonstration of how to: load multiple 2D live-imaging data files of the nuclear envelope; correct membrane segmentations and fluctuation results for each individual file; save the files of interest.

Journal of Cell Science • Supplementary information



Journal of Cell Science • Supplementary information

Figure S1. Trackosome User Interface: Centrosome Dynamics module. (A**)** Main menu to open the windows shown in B, C, D and E. (B) Load data window, where the user can import the 3D live-cell imaging videos from the centrosomes, nuclear membrane and cellular membrane (.tiff, .mat, .nd2). It is also possible to load .mat files previously exported by Trackosome ("Load Saved Data" button). These populate the entire interface with the data contained in the file, allowing the user to reexamine previous analysis. (C) Pre-processing window to filter the centrosomes channel, trim the videos, and select the approximate initial coordinates of the centrosomes. (D) Centrosomes tracking window. The user can follow the tracking results for each frame in real-time. The algorithm can be stopped at any time to correct eventual mistakes. There are two modes of coordinates correction: 1) "Change ROI": move and resize the regions-of-interest to guarantee that they contain the centrosomes (option currently selected in the image shown); 2) "Change Coords Manually": manual selection of the new 3D coordinates of a centrosome. The algorithm can proceed from the corrected frame. These corrections can also be done after analyzing the full video. The array of blue buttons at the bottom allows the user to navigate between frames and provide visual feedback regarding the tracking status of each frame: blue - ok; yellow - problem finding the centrosomes; red - forced break due to error; gray - coordinates manually changed. The user can also discard specific frames from the analysis in the "Remove Frames" section. (D') Window to change the main settings of the tracking algorithm. (E) Results window where the user can inspect and save the results obtained. The results can be exported as .xlsx, .csv files and also a .mat file that stores all the data from the interface. The .mat file allows the user to reload the full Trackosome interface with the stored data. Also, the user can access relevant stored variables (such as the membrane reconstructions) by loading this .mat file directly on the MATLAB command window. See Movie 1.

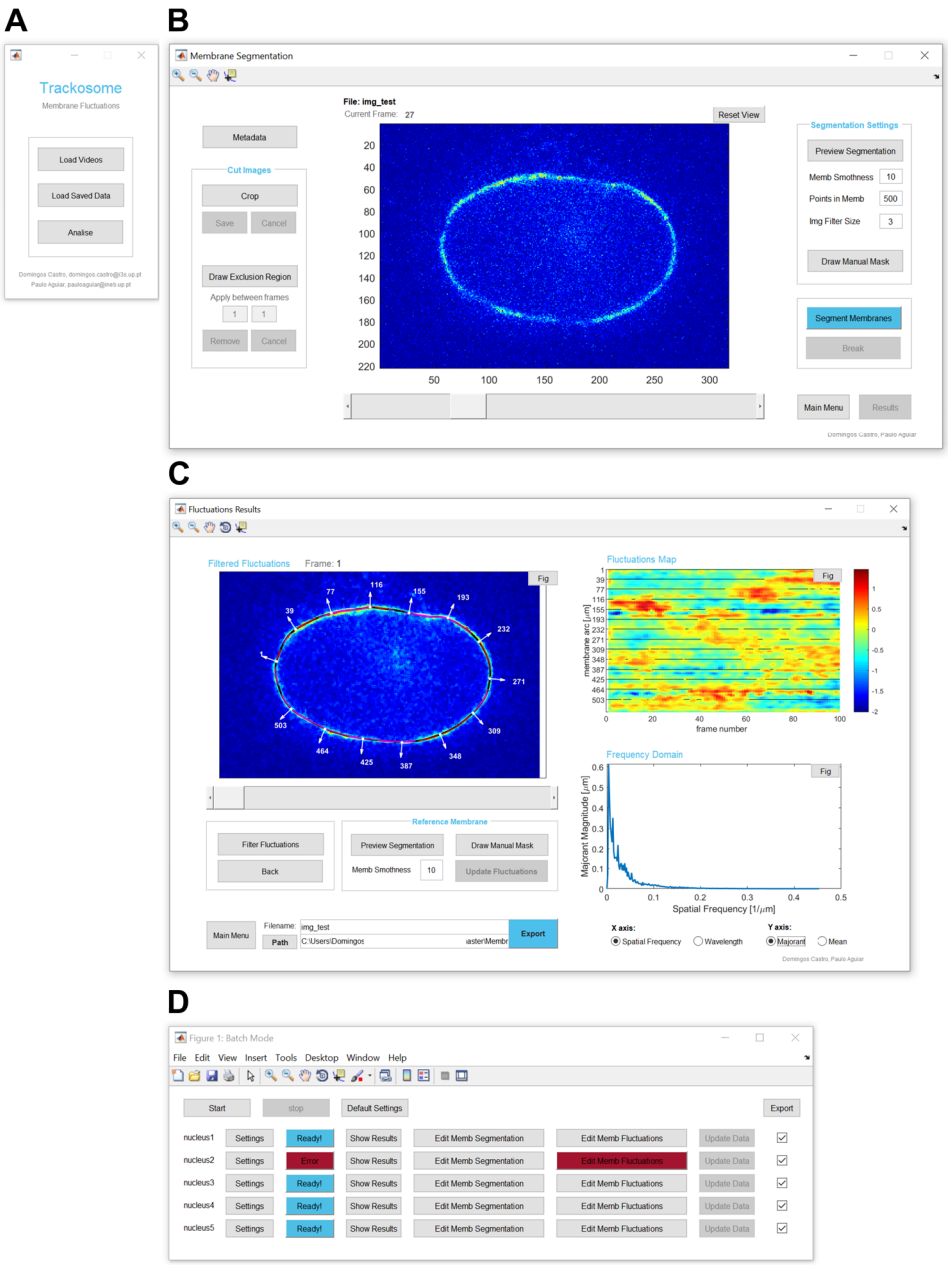


Figure S2. Trackosome User Interface: Membrane Fluctuations module. (A) Main menu to load the 2D videos (.tiff, .mat, .nd2). The loading function used is *uipickfiles* (Douglas Schwarz 2020). If only one video is selected, the user is directed to the window shown in B. If multiple files are selected, the user is directed to batch analysis mode, shown in D. It is also possible to open previously exported files and load the entire interface with the imported data. (B) Window to perform membrane segmentation. The user can edit the video before segmenting the membranes. The edit options include cropping the frames, drawing masks to guide membrane segmentation, and removing manually drawn regions from specified frames (to eliminate, for example, high intensity noise blobs located near the membrane). It is possible to preview the segmentation of any frame to optimize parameters before starting the segmentation of the entire video. (C) Results window where the user can inspect, correct and export the results obtained. The corrections include editing the reference membrane and adjusting the spatiotemporal filters

applied to the membrane fluctuations. (D) Window for batch analysis, opened if the user selects multiple videos in the Main Menu. In the window shown, five files were selected. Each file is associated with a set of buttons to edit parameters, show the results obtained, edit the membrane segmentation (opens window B), edit fluctuation results (opens window C), and activate/deactivate exporting.

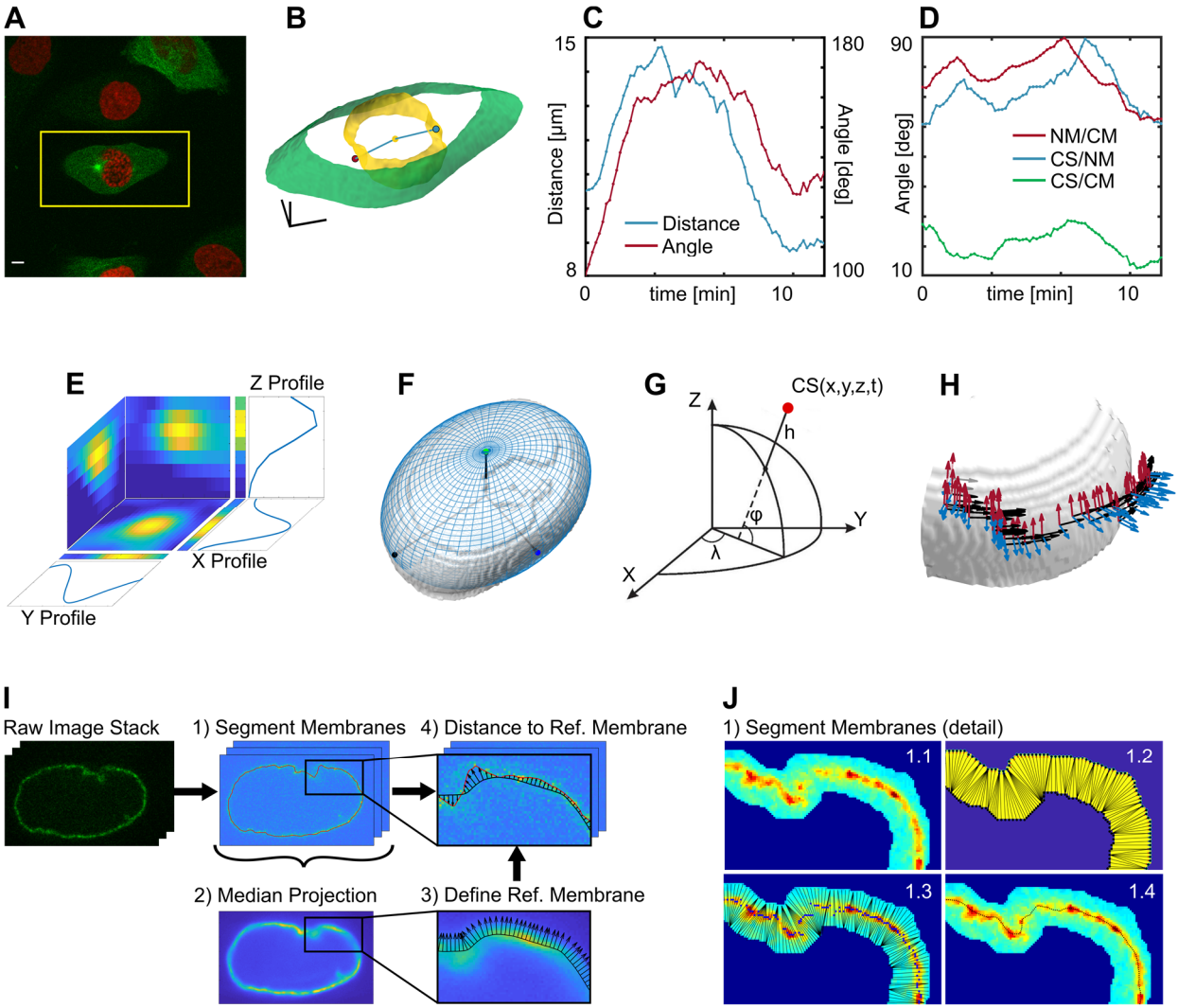


Figure S3. Details of Trackosome algorithms. (A-D) Analysis of unconstrained cells with Trackosome, to demonstrate how the algorithm can be used in multiple experimental conditions, and not only in micro-patterned cells. (A) Image of a RPE-1 cell expressing tubulin-GFP/H2B-RFP in a fibronectin coated glass coverslip. The centrosomes can be easily identified in green channel and the nucleus in the red channel. The green channel can also be used to identify the cellular membrane, because the cellular limit is clear enough for Trackosome to reconstruct it. The yellow rectangle marks the cropping performed in Trackosome to analyze a single cell - the nuclear and cellular membrane reconstruction algorithms require isolated cells. Scale-bar: 5 µm. (B) 3D reconstruction of the centrosomes, nuclear envelope and cellular membrane, for the frame shown in A. Scale-bars: 5 µm. (C) Distance (blue) and angles (red) between centrosomes. The distance between centrosomes starts decreasing once they form a 180° angle with the nucleus centroid, that is, once they are at opposite sides of the nucleus. (D) Angles formed between: centrosomes axis and the nucleus major axis (blue); centrosomes axis and the cell major axis (green); the nucleus major axis and the cell major axis (red). These curves show that the nucleus is displaced perpendicularly to the cellular membrane and the centrosomes are aligned with the cellular membrane, as shown in B. (E) Detail of the centrosome tracking algorithm (Centrosome

Dynamics module). Example of the initial region-of-interest (ROI) obtained for a centrosome at a given frame. The three planes represent the intensity sum projections of the ROI. The ROIs are centered in the coordinates of the centrosome obtained in the previous frame. The initial ROI is shortened until it confines a blob with the dimensions of the centrosome. The short ROI converges to the peaks of intensity shown in the X, Y, Z profiles. The final 3D coordinates of the centrosome are found by fitting a Gaussian curve to the intensity profiles along the X, Y and Z axes extracted from the shortened ROI. The (x,y,z) position of the centrosome corresponds to the mean value of each Gaussian curve. (F-H) Details of centrosome trajectories in ellipsoidal coordinates. (F) Ellipsoid (blue) fitted to the median nucleus (grey) used as referential for the ellipsoidal coordinates. (G) Conversion from Cartesian (x,y,z) to ellipsoidal (ϕ, λ, h) coordinates based on the referential system defined in F. The red dot represents a centrosome (CS) in a given time point (t). (H) Each point of the centrosomes trajectories is associated with an orthonormal basis defined by a latitude unit vector (red), longitude unit vector (black) and height unit vector (blue). These indicate the direction along which the ellipsoidal velocities and accelerations are calculated for each position. (I-J) Details of the membrane fluctuations algorithm (Nuclear Envelope Fluctuations module). (I) Overview of the four main steps of the algorithm: 1) membrane segmentation and centering; 2) median projection of the centered frames; 3) segmentation of the membrane obtained in 2, defining the reference membrane and the associated normal vectors; 4) Definition of the membrane fluctuations for each frame as the distance between the reference membrane and the membrane of the current frame, along the direction defined by the vectors orthogonal to the reference membrane. (J) Detail of the membrane segmentation step (1) for a given frame: 1.1) filter and mask the image; 1.2) the points that constitute the two borders of the mask are connected by the shortest segment; 1.3) find the pixel with maximum intensity for each segment defined in 1.2; 1.4) filter the positions of the pixels found in 1.3.

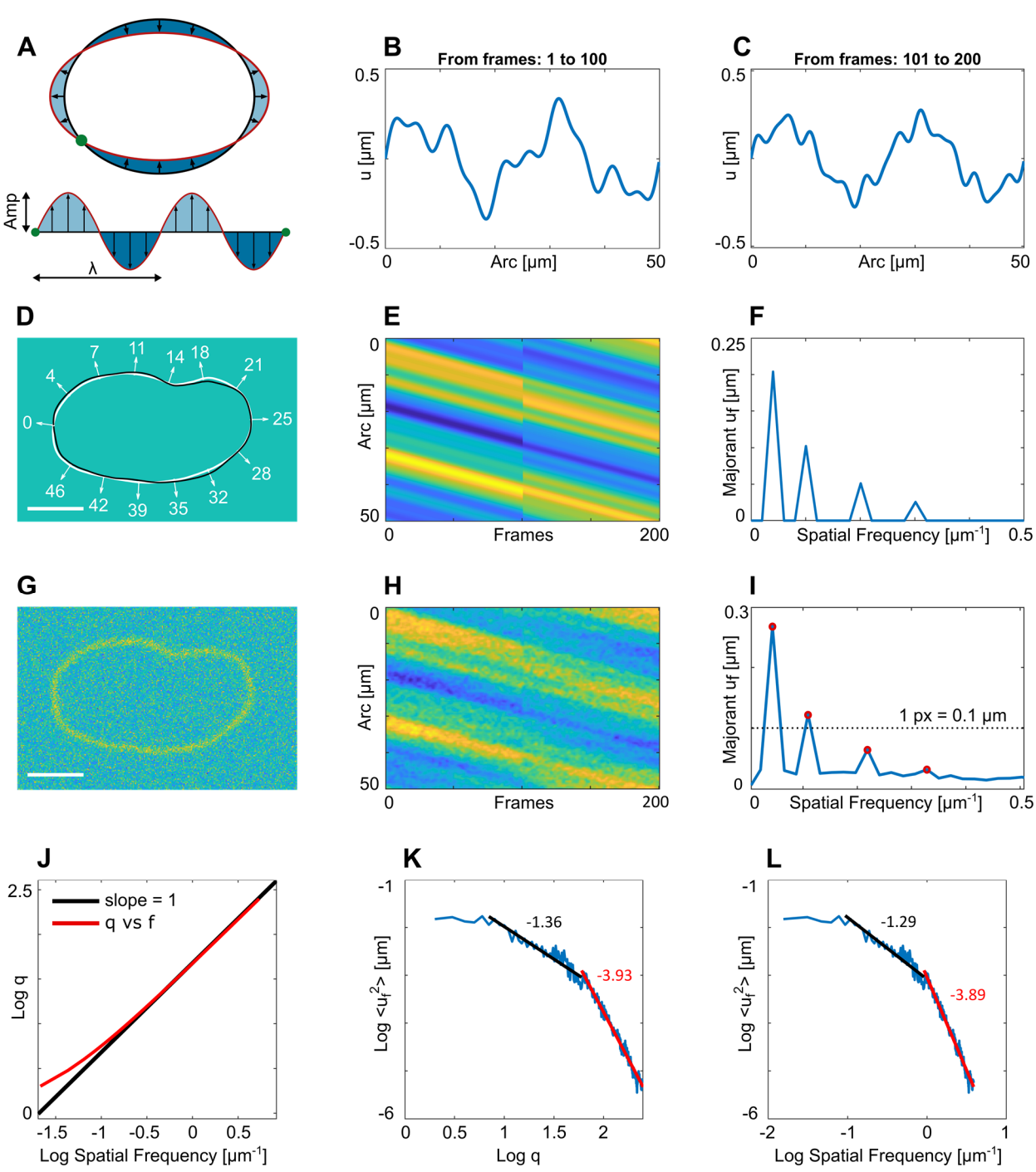


Figure S4. Validation of the membrane fluctuations. (A-I) Validation of the fluctuations values using synthetic videos. (A) Unfolding the distance between the current membrane (red) and the reference membrane (black) generates a waving signal (bottom) with characteristic wavelengths. For the compression fluctuation represented here, the associated unfolded fluctuations have a wavelength (λ) of half perimeter. (B) Unfolded fluctuation used between frames 1 and 100 to create the synthetic validation video. The full synthetic video has a total of four sinusoidal components: f_1 = 0.05 µm^{-1}, Amp_1 = 0.2 µm; f_2 = 0.1 µm^{-1}, Amp_2 = 0.1 µm; f_3 = 0.2 µm^{-1}, Amp_3 = 0.05 µm; f_4 = 0.3 µm^{-1}, Amp_4 = 0.025 µm. The fluctuations created between frames 1 and 100

have the components: {f_1, Amp_1}, {f_2, Amp_2} and {f_3, Amp_3}. (C) Unfolded fluctuation used between frames 101 and 200. The signal has three sinusoidal components: {f_1, Amp_1}, {f_3, Amp_3} and {f_4, Amp_4}. (D) First frame of synthetic membrane (black), waving around the reference membrane (white). Each frame was created by adding the associated fluctuation values to the reference membrane, along the direction of the normal vectors. Scale-bar: 5 µm. E) Expected fluctuation map of the entire video. The fluctuations (B and C) added to the reference membrane were continuously propagating around the nucleus in order to create a dynamic membrane. This generates the descending pattern shown in E. (F) Majorant of frequency depend fluctuations, u_f, obtained for the map E. The peaks of this curve reveal the amplitude and wavelengths of all the sinusoidal waves used to create the fluctuations. (G) First frame of the final synthetic video. Each frame was created by mapping the raw synthetic membranes (as shown in D) into a grid of pixels (1 px = 0.1 µm), followed by the addition of random noise. Scale-bar: 5 Scaler-bar: 5 µm. (H) Fluctuations map obtained by analyzing the noisy video G with Trackosome. The map is very similar to the expected result E, clearly showing the two different fluctuation patterns (before and after frame 100). (I) Majorant u_f obtained with Trackosome for the noisy video G, revealing the four expected peaks: $f_1^* = 0.044$ µm^{-1}, $Amp_1^* = 0.26$ µm; $f_2^* = 0.11$ µm^{-1}, $Amp_2^* = 0.12$ µm; $\lambda_3^* = 0.22$ µm^{-1}, $Amp_3^* = 0.065$ µm; $\lambda_4^* = 0.33$ µm^{-1}, $Amp_4^* = 0.032$ µm. Trackosome is able to detect fluctuations with amplitudes at the subpixel level in a highly noise environment. (J-K) Equivalence between slopes using wavenumber (q) and spatial frequency. (J) In logarithmic scales, q and spatial frequency are equivalent for frequencies above 0.1 µm^{-1}, making the slope analysis viable for the considered range. (K) Representative example of the average FT of the squared fluctuations, $\langle u_q^2 \rangle$, for a given cell, plotted against q, and the slopes of the linear fits (solid lines) at low and high frequencies. (L) Same $\langle u_f^2 \rangle$ curve as in B, but plotted against the associated spatial frequency. The slopes of the linear fits are very similar to those obtained in B.

References

Douglas Schwarz (2020). uipickfiles: uigetfile on steroids (https://www.mathworks.com/matlabcentral/fileexchange/10867-uipickfiles-uigetfile-on-steroids), MATLAB Central File Exchange. Retrieved October 15, 2020.