

Membrane Kymograph Generator: a cross-platform GUI software for automated generation and analysis of kymographs along dynamic cell boundaries

Tatsat Banerjee^{1,2,†,*,}, Bedri Abubaker-Sharif^{1,3,}, Peter N. Devreotes^{1,4,*,}, Pablo A. Iglesias^{1,3,5,*,}

¹Department of Cell Biology and Center for Cell Dynamics, School of Medicine, Johns Hopkins University, Baltimore, MD 21205, United States

²Department of Chemical and Biomolecular Engineering, Whiting School of Engineering, Johns Hopkins University, Baltimore, MD 21218, United States

³Department of Biomedical Engineering, Whiting School of Engineering and School of Medicine, Johns Hopkins University, Baltimore, MD 21205, United States

⁴Department of Biological Chemistry, School of Medicine, Johns Hopkins University, Baltimore, MD 21205, United States

⁵Department of Electrical and Computer Engineering, Whiting School of Engineering, Johns Hopkins University, Baltimore, MD 21218, United States

*Corresponding authors. Tatsat Banerjee, Department of Cell Biology and Center for Cell Dynamics, School of Medicine, Johns Hopkins University, Baltimore, MD 21205, United States. E-mail: tatsatb@jhu.edu; Peter N. Devreotes, Department of Cell Biology and Center for Cell Dynamics, School of Medicine, Johns Hopkins University, Baltimore, MD 21205, United States. E-mail: pnd@jhmi.edu; Pablo A. Iglesias, Department of Electrical and Computer Engineering, Whiting School of Engineering, Johns Hopkins University, Baltimore, MD 21218, United States. E-mail: pi@jhu.edu.

[†]Present address: Howard Hughes Medical Institute, Whitehead Institute for Biomedical Research, Massachusetts Institute of Technology, Cambridge, MA 02142, United States

Associate Editor: Virginie Uhlmann

Abstract

Summary: The plasma membrane and accompanying cortex serve as major hubs of signal transduction and cytoskeletal activities that collectively regulate cell physiological processes such as migration, polarity, macropinocytosis, phagocytosis, and cytokinesis. Yet, dynamically tracking membrane-cortex associated protein or lipid kinetics from live-cell image series remains challenging, primarily due to the difficulty of accurately extracting and aligning the cell boundary between consecutive frames as the cell continuously deforms and moves. Here, we present *Membrane Kymograph Generator*, a cross-platform software that accepts multichannel time-lapse live-cell fluorescent imaging datasets and automates boundary tracking, inter-frame alignment, and intensity sampling along the boundary. The software implements a rotational offset minimization algorithm that aligns boundaries across consecutive frames by exhaustively searching for the optimal angular shift that minimizes point-to-point distances, while handling variations in boundary point counts due to cell shape changes. The software outputs kymographs representing spatiotemporal dynamics of membrane-associated proteins or biosensors, allows users to fine-tune visualization parameters through an interactive interface, and provides built-in correlation analysis tools for multi-channel datasets. Furthermore, a native Python API enables programmatic usage for batch processing and further downstream analysis. Validation tests demonstrated that the *Membrane Kymograph Generator* accurately tracks, visualizes, and quantitates the spatial kinetics of a wide array of membrane proteins and lipid biosensors over extended time periods, in a variety of cell types including *Dictyostelium* amoeba, human neutrophils, mouse macrophages, and mammalian cancer cells. The GUI-based software is user-friendly, requires no technical expertise, and significantly reduces the manual effort required for kymograph generation and analysis while ensuring high accuracy and reproducibility.

Availability and Implementation: Membrane Kymograph Generator is free and open-source, licensed under GNU General Public License 3.0 or later. It can be installed on both x86-64 and AArch64/ARM64 computers running Windows, macOS, or any standard Linux distribution. The software is distributed as standalone installer files and portable executables targeting specific architectures and operating systems, requiring no dependency resolution. The source code, documentation/wiki, installers, portable binaries, and test data are freely available at <https://github.com/tatsatb/membrane-kymograph-generator>. The software can also be installed via PIP (package ID: *membrane-kymograph*, <https://pypi.org/project/membrane-kymograph>) and accessed programmatically via a built-in Python API. The source code is also archived on Zenodo (DOI: 10.5281/zenodo.20318834).

Received: 12 February 2026. Revised: 27 May 2026. Accepted: 10 June 2026

© The Author(s) 2026. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

The plasma membrane and associated cortex undergo rapid remodeling driven by coordinated activities of lipid and protein components that collectively define signal transduction and cytoskeletal organizations, and thereby regulate cell physiological processes such as migration, polarity, macropinocytosis, phagocytosis, and cytokinesis (Ridley *et al.* 2003, Swaney *et al.* 2010, Banerjee *et al.* 2025). Advances in live-cell fluorescence microscopy and genetically-encoded biosensors have enabled monitoring of membrane dynamics with high spatiotemporal resolution, yet quantitative analysis remains challenging due to the curved, complex geometry of cell membranes and the difficulty of decoupling morphological changes from protein or lipid dynamics.

Kymographs, two-dimensional space-time plots representing spatial position and fluorescence intensity over time, have emerged as powerful visualization tools in cell biology and biophysics. However, generating accurate membrane kymographs from time-lapse datasets remains non-trivial, as cell boundaries undergo large-scale deformations during dynamic processes. While deep learning-based tools such as *ilastik* (Berg *et al.* 2019), *CellPose* (Stringer *et al.* 2021), *CellTracker* (Hu *et al.* 2021), *CellDetective* (Torro *et al.* 2025), *DeepCell* (Greenwald *et al.* 2022), and *CellProfiler* (Kamentsky *et al.* 2011, Stirling *et al.* 2021) can accurately segment cell boundaries, they do not track and align boundaries that assume different sizes and geometries between frames. Existing Fiji/ImageJ (Schindelin *et al.* 2012) kymograph plugins such as *KymographBuilder* (Mary *et al.* 2016) and *Multi Kymograph* (https://github.com/fiji/Multi_Kymograph) (Rietdorf and Seitz 2017) only work with static, user-defined lines. Consequently, researchers have relied on manual or semi-automated approaches, analyzed straight-line membrane segments, or developed custom code: methods that are time-consuming, prone to user-bias, and often inaccessible to experimental biologists (Bosgraaf *et al.* 2009, Arai *et al.* 2010, Bosgraaf and Van Haastert 2010, Tyson *et al.* 2010, Xiong and Iglesias 2010, Barry *et al.* 2015, Miao *et al.* 2017, van Haastert *et al.* 2017, Baniukiewicz *et al.* 2018, Cheng and Mullins 2020, Lee *et al.* 2020, Ghabache *et al.* 2021, Banerjee *et al.* 2022, 2023, Pal *et al.* 2023, Tong *et al.* 2023, Chua *et al.* 2024, Gong *et al.* 2024, Lin *et al.* 2024, Deng *et al.* 2025, Iwamoto *et al.* 2025, Kuhn *et al.* 2025). Additionally, some of these tools require user registrations or were not released under an approved free/libre license (<https://www.gnu.org/licenses/license-list.html>) (Free Software Foundation 2025). To our knowledge, no free and open-source standalone software fully automates membrane kymograph generation while accounting for dynamically deforming cell boundaries.

Here, we present *Membrane Kymograph Generator*, a GNU GPL v3 licensed, cross-platform GUI software that automates the entire process of generating and analyzing kymographs along dynamic boundaries from multichannel time-lapse live-cell fluorescent images. The software extracts cell boundaries, tracks and aligns them across frames using a rotational offset minimization algorithm, samples fluorescence intensities along the boundary, applies LOWESS smoothing and normalization, exports outputs in various accessible formats, generates publication-quality kymographs for each channel, and provides

built-in correlation analysis tools for multi-channel datasets. A Python API enables programmatic access, batch processing, and further downstream analysis. Importantly, unlike previous tools that require knowledge in MATLAB/Python/Julia for parameter tuning, *Membrane Kymograph Generator* provides an intuitive interface suitable for researchers with any level of computational expertise.

2 Software architecture and results

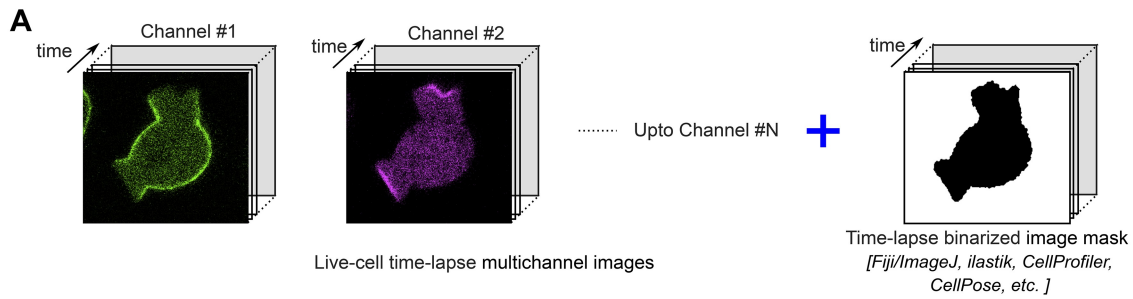
Membrane Kymograph Generator is implemented as a Python package with additional components in bash and Inno Setup scripts. The software uses NumPy, SciPy, and Pandas for numerical operations; scikit-image, OpenCV, Shapely, and Pillow for image processing; Matplotlib and Seaborn for visualization; joblib for parallel processing; and ttkbootstrap for the graphical user interface. The modular architecture supports both GUI-based and programmatic workflows via a built-in Python API (Fig. 1).

2.1 Installation and updates

The software can be installed on computers with x86-64 and ARM64 architectures, running Windows, macOS, or Linux. It is primarily distributed as standalone installable packages (.exe for Windows, .dmg for macOS, .appimage for Linux). Additionally, portable binary executables and tarballs for each platform are also provided. The codebase has been tested on Windows 10/11, Ubuntu 22.04/24.04 LTS, and macOS 15. For programmatic access, the software can also be installed inside any standard Python environment using PIP from the Python Package Index (PyPI) repository (package ID: *membrane-kymograph*) or built from source. A GitHub Actions-based CI/CD pipeline automatically builds, tests, and deploys new versions upon tagged commits to the main branch of the GitHub repository.

2.2 Graphical user interface overview and input files

The GUI, built using ttkbootstrap/tkinter, provides an intuitive interface for general users (Fig. 1B–E). The main interface features organized panels where users select their multi-channel time-lapse image and binarized mask files via native file browser dialogs. Users configure key parameters including membrane width for perpendicular intensity sampling (default: 8 pixels, adjustable), number of fluorescence channels (1–10), colormap (default modified Parula, plus viridis, plasma, inferno, gnuplot, and others), and output formats (PNG, SVG, PDF). A progress panel provides real-time feedback, while a preview panel displays tracked boundaries with perpendicular sampling lines overlaid on fluorescence images, with an integrated frame slider for inspecting tracking quality across the time series. The preview panel also shows the generated kymographs after processing completes. Auxiliary dialog windows allow interactive fine-tuning of intensity scaling and colormap for previously generated kymographs without reprocessing (“Adjust Existing Kymograph”), and frame-by-frame Pearson and Spearman



B

Membrane Kymograph Generator
[Graphically installable across x86-64 and ARM64 devices, running Windows, macOS, or any Linux distro]

Main GUI Window

Adjust Kymograph Color Limits

Load Kymograph File

Kymograph (npy):

Color Limit Controls

Auto Suggest Limits

Lower Limit: Upper Limit:

Colormap:

Kymograph Preview

Kymograph Preview

Normalized Intensity

Frame

Kymograph Correlation Analysis

Load Kymograph File

Kymograph 1 (npy):

Name: PTEN

Kymograph 2 (npy):

Name: PHCrc

Analysis Options

Exclude zero values from correlation

Note: Zero values and both values will be excluded from analysis.

Status

Ready. Select two kymograph files to analyze.

Starting correlation analysis

Kymograph 1: kymo_channel_1_smooth.npy
Kymograph 2: kymo_channel_2_smooth.npy
Kymograph 1 File:

Output directory: home/tatsab/Python_Files/kymoexample/Kymo_Processed_Data/Correlation_Analysis

Analysis completed

- Processed 42 frames
- Created file: correlation_analysis.xlsx
- Created plot: correlation_coefficients.png
- Created plot: correlation_pvalues.png
- Created plot: correlation_pvalues.png

Summary Statistics

Pearson R: mean=0.536, std=0.188

Spearman R: mean=0.525, std=0.206

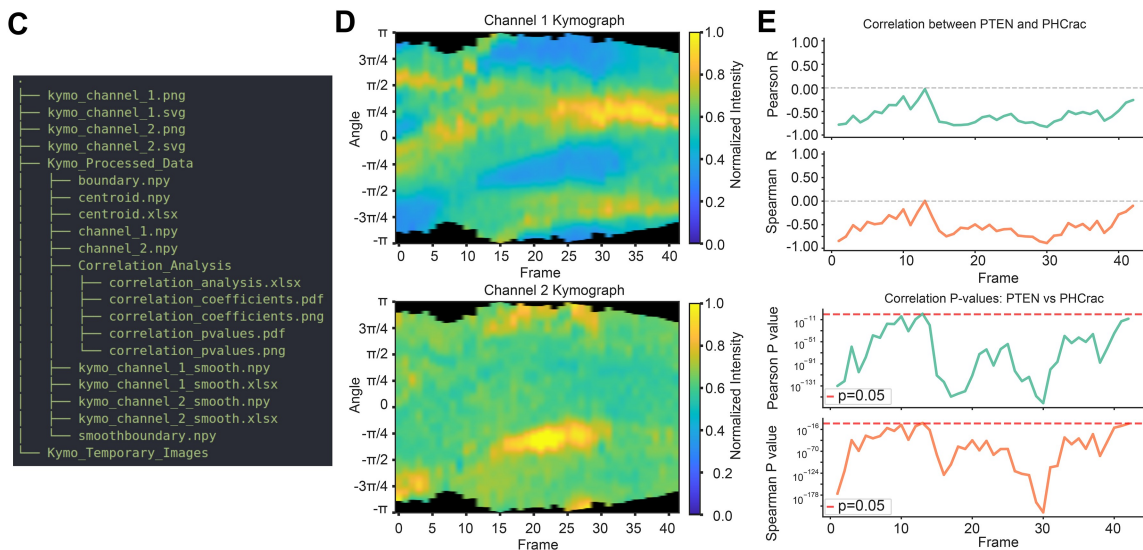


Figure 1 Overview of the default GUI workflow for Membrane Kymograph Generator. (A) Input files for the Membrane Kymograph Generator include a multichannel time-lapse live-cell imaging dataset and a binarized mask of the whole cell. Note that the binarized mask can be generated using various standard tools such as Fiji/ImageJ, ilastik, CellProfiler, CellPose, etc., or using custom code involving thresholding, morphological operations, or machine learning-based approaches. (B) Graphical User Interface of Membrane Kymograph Generator. The main interface is shown on the left whereas auxiliary windows are shown on the right. The main interface allows users to upload their input files, configure analysis parameters, and

correlation analysis between channels with significance testing (“Correlation Analysis”; Fig. 1E). Custom configuration files can be saved and loaded for consistent parameters across datasets. The software accepts standard multi-frame TIFF files (Fig. 1A), with multichannel files processed at once by specifying the channel count. All outputs are organized into structured directories (Fig. 1C), facilitating reproducibility and downstream analysis.

2.3 Membrane coordinate extraction

The software begins by loading the user-provided binarized mask, which can be generated using standard segmentation tools such as Fiji/ImageJ, ilastik, CellProfiler, CellPose, or custom approaches. For each frame, the cell boundary is identified using contour detection (`skimage.measure.find_contours`) (van der Walt *et al.* 2014), the largest enclosed area is selected, and the geometric centroid is computed. Boundary coordinates are interpolated with a piecewise cubic Hermite interpolating polynomial (PCHIP) (Fritsch and Carlson 1980), which preserves shape and prevents oscillations, and then resampled to uniform arc-length spacing to distribute boundary points evenly along the contour.

2.4 Inter-frame boundary alignment

Cell boundaries undergo substantial shape changes and rotations between consecutive frames due to dynamic cellular processes. The software addresses this through a rotational offset minimization algorithm that aligns boundaries across frames by exhaustively searching for the optimal angular shift. For the initial frame, the boundary is rotationally aligned to a specified angular position (default: 180 degrees) by calculating polar angles relative to the centroid and circularly shifting the boundary array. For subsequent frames, the algorithm tests all possible circular shifts relative to the previous frame’s aligned boundary. To handle varying boundary point counts, the longer boundary is subsampled by randomly excluding points until both match in size; the mean Euclidean distance between corresponding points is then computed for each shift, and the minimum-distance shift is selected. This approach ensures that homologous membrane regions are consistently tracked despite large-scale morphological changes. Aligned boundaries serve as references for the next frame, creating a temporally consistent coordinate system throughout the time series. Notably, this exhaustive search imposes no hard restriction on the inter-frame interval, although ultra-long intervals relative to the cell’s

membrane dynamics may yield occasional vertical discontinuities when two membrane patches produce nearly identical alignment costs.

2.5 Perpendicular intensity sampling

After alignment, fluorescence intensities are sampled perpendicular to the membrane at each boundary point. For each point, a local circular arc is fit to a small neighborhood using least-squares circle fitting (`circle_fit.standardLSQ`; <https://github.com/AlliedToasters/circle-fit>) (Klear and Lauritzen 2023), providing a robust estimate of local curvature and normal direction. The perpendicular sampling direction is determined from the vector connecting the fitted circle center to the boundary point. Intensities are sampled at 0.5-pixel intervals extending both inward and outward from the boundary (default: ± 8 pixels, adjustable via `l_perp`), using biquadratic interpolation (`scipy.ndimage.map_coordinates` with `order=2`) (Virtanen *et al.* 2020). To reduce noise, the mean of the top 5 intensity values within each perpendicular profile is used. This process is parallelized using `joblib`, accelerating processing for long boundaries. The resulting intensity profiles capture both the membrane-localized signal and the immediate cortical region, enabling quantification of proteins and lipids enriched at or near the plasma membrane. These intensity profiles form the basis for subsequent kymograph generation and analysis. The fidelity of these profiles depends on the upstream mask quality: while small mask offsets are tolerated by the $\pm l_perp$ sampling window, heavily over-smoothed masks may miss fine features such as filopodia or sharp invaginations.

2.6 Kymograph generation and normalization

Intensity profiles from all boundary points and frames are organized into kymograph matrices (boundary position \times time). A locally weighted scatterplot smoothing (LOWESS) algorithm (`statsmodels.nonparametric.smoothers_lowess`) (Cleveland 1979) is applied along the spatial dimension for each frame with a smoothing fraction of 0.1, filtering high-frequency variations while preserving genuine features. The kymograph is then normalized using the 2nd and 98th percentiles of the raw intensity distribution, linearly rescaled to [0, 1], and clipped, ensuring consistent visualization across channels and conditions. This normalization strategy is particularly effective for live-cell fluorescence data, where photobleaching and differences in expression levels can complicate quantitative comparisons.

Figure 1. Continued

visualize the progress and error messages (if any) in real-time, and finally visualize the generated kymographs. The bottommost panel here shows the tracked boundaries and intensity sampling domain. The top right auxiliary window allows the users to adjust the visualization settings, such as colormap and intensity scaling. The bottom right auxiliary window facilitates the downstream correlation analysis across different kymographs. (C) List of key output files generated by the Membrane Kymograph Generator. The temporary images inside the *Kymo_Temporary_Images* folder are hidden from the tree view here. (D) Representative output kymographs generated by Membrane Kymograph Generator from two-channel live-cell images over 42 frames. Evidently, each column corresponds to a specific time point, and each row corresponds to a specific position along the cell boundary. The intensity values in the kymographs indicate the fluorescence intensity of the respective proteins at each position and time point. The colormap used is shown on the right. (E) Representative output correlation analysis plots generated by the “Kymograph correlation Analysis” module of Membrane Kymograph Generator. The top two plots show the Pearson’s and Spearman correlation coefficient (R) values between two kymographs over time, indicating the degree of correlation between the two proteins’ dynamics along the membrane boundary. The bottom two plots show respective *P* value associated with the correlation coefficients, indicating the statistical significance of the observed correlations.

Normalized kymographs are saved as NumPy files (.npy) and Excel spreadsheets (.xlsx).

2.7 Visualization and export

The software generates publication-quality kymograph visualizations with customizable colormaps, including the default Parula colormap (adapted from MATLAB; <https://www.mathworks.com>) (MathWorks 2025). Intensity scaling can be adjusted interactively. Kymographs are exported as high-resolution PNG (300 DPI), SVG, and PDF, alongside raw data matrices (.npy and .xlsx). Preview images showing tracked boundaries with sampling lines are generated for verification. All outputs are organized in structured directories (Kymo_Processed_Data and Kymo_Temporary_Images).

2.8 Multi-channel correlation analysis

For multi-channel datasets, a built-in correlation module quantifies spatiotemporal relationships between membrane-associated proteins or biosensors. The software computes both Pearson and Spearman correlations between two kymographs on a frame-by-frame basis, with optional exclusion of zero-valued pixels to avoid background artifacts. Statistical significance is assessed via *P*value, and the software generates time series plots of correlation coefficients and *P*value with significance thresholds (Fig. 1E). All results are exported as Excel spreadsheets and publication-quality plots (PNG and PDF), facilitating investigation of co-localization dynamics and functional coupling between signaling components.

2.9 Python API, batch processing, and further analysis

Beyond the GUI, the software provides a Python API (`membrane_kymograph`) for programmatic access and batch processing. Users instantiate a `KymographProcessor` object and call its `process()` method with image/mask paths and parameters. The modular architecture allows advanced users to access intermediate processing steps, such as boundary extraction, alignment, and intensity sampling, for custom analyses. The API also supports parallel processing through Python's multiprocessing capabilities. The *GitHub Wiki* provides comprehensive documentation with example scripts for common batch processing patterns, including handling complex parameter configurations via CSV files, interrupted workflow resumption, and intensity profiling at specific spatiotemporal locations. The exported NumPy arrays can be readily imported into scientific Python libraries for advanced analyses including feature extraction, clustering, and predictive modeling, extending the utility of the software beyond interactive single-cell analyses.

2.10 Dissecting the dynamics of Ras/PI3K/F-actin network in migrating cells using the software

We validated the software using microscopy images from cells across the phylogenetic tree with diverse membrane-associated

proteins/lipids, in different physiological conditions. Combining GUI and API modes, >100 cells have been processed as they were migrating or making macropinosomes. As a proof of concept, here we present data from *Dictyostelium* amoeba, human neutrophils, and murine macrophages, collectively dissecting the evolutionarily conserved Ras/PI3K/F-actin network during amoeboid migration under both stochastic and receptor-driven activation. A key aspect of migration is symmetry breaking at the membrane-cortex as specific components asymmetrically localize to front-state/protrusion or back/basal-state regions (Ridley *et al.* 2003, Banerjee *et al.* 2025). These asymmetric distributions play key roles in organizing cellular polarity and regulating migration. Our kymographs and correlation analysis show that PTEN and PI(3,4,5)P3 biosensor PH_{Crac} consistently maintain tight complementarity in migrating *Dictyostelium* cells (Fig. 1D and E; Fig. S1A). In HL-60 neutrophils, newly-polymerized F-actin biosensor Lifeact shows phases of activation and extinction during migration (Fig. S1B and C): Lifeact remained largely absent during contractions but showed strong membrane recruitment during protrusions. Ras activation biosensor Raf_{RBD} and PH_{Crac} show strong positive correlation with slight phase difference (Fig. S1D–F), indicating Ras positively regulates PI3K at the leading edge. We also studied receptor-driven activation using RAW 264.7 macrophages: upon global C5aR activation, PI(3,4,5)P3 biosensor PH_{Akt} shows robust membrane recruitment (Fig. S1G–I), with temporal profiles showing sharp increase post-stimulation followed by adaptation (Fig. S1I), consistent with known dynamics of C5aR-PI3K signaling axis. Together, all these observations validate the accuracy and reliability of our software as these are consistent with previous findings in the field that delineated the roles of internal feedback loops in regulation of membrane-associated signaling network dynamics.

3 Discussion

We have developed *Membrane Kymograph Generator*, a free, open-source, cross-platform software that addresses a significant gap in quantitative live-cell imaging analysis by fully automating kymograph generation and downstream analysis along dynamic cell boundaries. This software removes substantial technical barriers that have historically limited membrane dynamics analysis to specialized computational researchers or forced reliance on time-consuming, manual/error-prone approaches. The modular Python architecture features both an accessible GUI for routine use and a Python API for programmatic access, accommodating users from experimentalists seeking straightforward kymograph generation to computational researchers requiring batch processing and custom analysis pipelines.

We view *Membrane Kymograph Generator* as an important addition to the broader ecosystem of bioimage analysis tools and users should choose according to their specific scientific question. Static, user-defined line kymographs remain best served by *KymographBuilder* and *Multi Kymograph* in Fiji/ImageJ. For quantifying intracellular transport in neuronal projections, *KymoAnalyzer* is a suitable option (Neumann *et al.* 2017). Specific whole-cell morphodynamics analyses such as edge

velocity maps or protrusion-retraction cycles can be handled by tools like *Quimp* (Bosgraaf and Van Haastert 2010) and related platforms. Upstream cell segmentation and tracking tools, as we mentioned earlier, can complement rather than compete with our software; a typical workflow uses one of these tools to produce the binary mask that *Membrane Kymograph Generator* then uses. *Membrane Kymograph Generator* is the natural choice when the question is specifically about the spatiotemporal dynamics of membrane- or cortex-localized proteins and lipid biosensors along the entire deforming cell boundary, when reproducibility and accessibility to non-programming users are priorities, or when batch processing across many cells is required.

At present, the software works as a native desktop application; in future, a cloud-native version could enable analysis of very large datasets and collaborative sharing. The software currently processes only one cell per binary mask (the largest connected component is selected when multiple are present); future versions may support multiple labeled cells from a single mask. Additional quantitative modules, such as automated spatiotemporal feature detection, frequency analysis of membrane oscillations, and machine learning-based feature extraction, could also be integrated; currently, such analyses require custom programs using our Python API.

In summary, *Membrane Kymograph Generator* provides the cell biology and biophysics communities with a tool for automated, reproducible, and quantitative analysis of membrane dynamics from live-cell imaging data. Its combination of robust algorithms, user-friendly design, cross-platform availability, and open-source licensing makes it accessible to a broad range of researchers and a valuable addition to the ecosystem of open-source bioimage analysis tools. We anticipate that this software will facilitate new discoveries delineating novel spatiotemporal regulation of membrane-associated signaling and cytoskeletal processes across diverse cellular systems and physiological contexts.

Acknowledgements

We thank all the members of the Iglesias and Devreotes laboratories (Johns Hopkins University) for their helpful feedback. We especially thank D. Biswas (Iglesias laboratory) for his contribution during the development of this software. We would also like to thank Y. Deng, D. S. Pal, and S. Ye (Devreotes laboratory) for their help during testing the software.

Author contributions

Tatsat Banerjee (Conceptualization [Lead], Data curation [Lead], Formal analysis [Lead], Investigation [Lead], Methodology [Lead], Project administration [Lead], Resources [Lead], Software [Lead], Supervision [Equal], Validation [Lead], Visualization [Lead], Writing—original draft [Lead], Writing—review & editing [Lead]), Bedri Abubaker-Sharif (Formal analysis [Supporting], Methodology [Supporting], Resources [Supporting], Validation [Supporting]), Peter N. Devreotes (Conceptualization [Supporting], Funding acquisition [Lead], Resources [Lead], Software [Lead], Supervision [Equal],

Validation [Equal], Writing—review & editing [Equal]), and Pablo A. Iglesias (Conceptualization [Lead], Formal analysis [Lead], Funding acquisition [Lead], Investigation [Supporting], Methodology [Equal], Project administration [Equal], Resources [Lead], Software [Lead], Supervision [Equal], Validation [Equal], Visualization [Equal], Writing—original draft [Supporting], Writing—review & editing [Equal])

Supplementary material

Supplementary material is available at *Bioinformatics* online.

Conflict of interests

None declared.

Funding

This work was supported by the National Institutes of Health [R35 GM118177 to P.N.D., R01 GM149073 to P.A.I., S10 OD016374 to S. Kuo of the JHU Microscope Facility]; and the Air Force Office of Scientific Research [MURI FA95501610052 to P.N.D.]. This manuscript is the result of funding in whole or in part by the National Institutes of Health (NIH). It is subject to the NIH Public Access Policy. Through acceptance of this federal funding, NIH has been given a right to make this manuscript publicly available in PubMed Central upon the Official Date of Publication, as defined by NIH.

Data availability

All data needed to evaluate the conclusions of the paper are present in the main text or in the supplementary materials or in the GitHub repo (<https://github.com/tatsatb/membrane-kymograph-generator>). Any additional requests for information or data will be fulfilled by the corresponding authors upon reasonable request.

References

- Arai Y, Shibata T, Matsuoka S *et al.* Self-organization of the phosphatidylinositol lipids signaling system for random cell migration. *Proc Natl Acad Sci USA* 2010;**107**:12399–404.
- Banerjee T, Biswas D, Pal DS *et al.* Spatiotemporal dynamics of membrane surface charge regulates cell polarity and migration. *Nat Cell Biol* 2022;**24**:1499–515.
- Banerjee T, Deng Y, Pal DS *et al.* Signaling and actin waves at a glance. *J Cell Sci* 2025;**138**:jcs263634.
- Banerjee T, Matsuoka S, Biswas D *et al.* A dynamic partitioning mechanism polarizes membrane protein distribution. *Nat Commun* 2023;**14**:7909.
- Baniukiewicz P, Collier S, Bretschneider T. Quimp: analyzing transmembrane signalling in highly deformable cells. *Bioinformatics* 2018;**34**:2695–7.

- Barry DJ, Durkin CH, Abella JV *et al.* Open source software for quantification of cell migration, protrusions, and fluorescence intensities. *J Cell Biol* 2015;**209**:163–80.
- Berg S, Kutra D, Kroeger T *et al.* Ilastik: interactive machine learning for (bio)image analysis. *Nat Methods* 2019;**16**:1226–32.
- Bosgraaf L, Van Haastert PJM. Quimp3, an automated pseudopod-tracking algorithm. *Cell Adh Migr* 2010;**4**:46–55.
- Bosgraaf L, van Haastert PJM, Bretschneider T. Analysis of cell movement by simultaneous quantification of local membrane displacement and fluorescent intensities using Quimp2. *Cell Motil Cytoskeleton* 2009;**66**:156–65.
- Cheng KW, Mullins RD. Initiation and disassembly of filopodia tip complexes containing vasp and lamellipodin. *Mol Biol Cell* 2020;**31**:2021–34.
- Chua XL, Tong CS, Su M *et al.* Competition and synergy of Arp2/3 and formins in nucleating actin waves. *Cell Rep* 2024;**43**:114423.
- Cleveland WS. Robust locally weighted regression and smoothing scatterplots. *J Am Stat Assoc* 1979;**74**:829–36. <https://doi.org/10.1080/01621459.1979.10481038>
- Deng Y, Banerjee T, Matsuoka S *et al.* PIP5K-Ras bistability triggers plasma membrane symmetry breaking to define cellular polarity and regulate migration. bioRxiv, 2024.09.15.613115, <https://doi.org/10.1101/2024.09.15.613115>, 2025, preprint: not peer reviewed.
- Free Software Foundation. Various licenses and comments about them. 2025.
- Fritsch FN, Carlson RE. Monotone piecewise cubic interpolation. *SIAM J Numer Anal* 1980;**17**:238–46. <https://doi.org/10.1137/0717021>
- Ghabache E, Cao Y, Miao Y *et al.* Coupling traction force patterns and actomyosin wave dynamics reveals mechanics of cell motion. *Mol Syst Biol* 2021;**17**:e10505.
- Gong B, Johnston JD, Thiemicke A *et al.* Endoplasmic reticulum-plasma membrane contact gradients direct cell migration. *Nature* 2024;**631**:415–23.
- Greenwald NF, Miller G, Moen E *et al.* Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. *Nat Biotechnol* 2022;**40**:555–65.
- Hu T, Xu S, Wei L *et al.* CellTracker: an automated toolbox for single-cell segmentation and tracking of time-lapse microscopy images. *Bioinformatics* 2021;**37**:285–7.
- Iwamoto K, Matsuoka S, Ueda M. Excitable Ras dynamics-based screens reveal RasGEF is required for macropinocytosis and random cell migration. *Nat Commun* 2025;**16**:117.
- Kamentsky L, Jones TR, Fraser A *et al.* Improved structure, function and compatibility for CellProfiler: modular high-throughput image analysis software. *Bioinformatics* 2011;**27**:1179–80.
- Klear M, Lauritzen M. circle-fit: Release 0.2.1, May 2023.
- Kuhn J, Banerjee P, Haye A *et al.* Complementary cytoskeletal feedback loops control signal transduction excitability and cell polarity. *Nat Commun* 2025;**16**:7482.
- Lee RM, Campanello L, Hourwitz MJ *et al.* Quantifying topography-guided actin dynamics across scales using optical flow. *Mol Biol Cell* 2020;**31**:1753–64.
- Lin Y, Pal DS, Banerjee P *et al.* Ras suppression potentiates rear actomyosin contractility-driven cell polarization and migration. *Nat Cell Biol* 2024;**26**:1062–76.
- Mary H, Rueden C, Ferreira T. Kymographbuilder: Release 1.2.4, June 2016.
- MathWorks. MATLAB version: 25.1.0 (R2025a), 2025.
- Miao Y, Bhattacharya S, Edwards M *et al.* Altering the threshold of an excitable signal transduction network changes cell migratory modes. *Nat Cell Biol* 2017;**19**:329–40.
- Neumann S, Chassefeyre R, Campbell GE *et al.* KymoAnalyzer: a software tool for the quantitative analysis of intracellular transport in neurons. *Traffic* 2017;**18**:71–88.
- Pal DS, Banerjee T, Lin Y *et al.* Actuation of single downstream nodes in growth factor network steers immune cell migration. *Dev Cell* 2023;**58**:1170–88.e7.
- Ridley AJ, Schwartz MA, Burridge K *et al.* Cell migration: integrating signals from front to back. *Science* 2003;**302**:1704–9.
- Rietdorf J, Seitz A. Multi_Kymograph: Release 3.0.1, April 2017.
- Schindelin J, Arganda-Carreras I, Frise E *et al.* Fiji: an open-source platform for biological-image analysis. *Nat Methods* 2012;**9**:676–82.
- Stirling DR, Swain-Bowden MJ, Lucas AM *et al.* CellProfiler 4: improvements in speed, utility and usability. *BMC Bioinformatics* 2021;**22**:433.
- Stringer C, Wang T, Michaelos M *et al.* Cellpose: a generalist algorithm for cellular segmentation. *Nat Methods* 2021;**18**:100–6.
- Swaney KF, Huang C-H, Devreotes PN. Eukaryotic chemotaxis: a network of signaling pathways controls motility, directional sensing, and polarity. *Annu Rev Biophys* 2010;**39**:265–89.
- Tong CS, Xü XJ, Wu M. Periodicity, mixed-mode oscillations, and multiple timescales in a phosphoinositide-Rho GTPase network. *Cell Rep* 2023;**42**:112857.
- Torro R, Díaz-Bello B, Arawi DE *et al.* Celldetective: an AI-enhanced image analysis tool for unraveling dynamic cell interactions. *eLife* 2025. <https://doi.org/10.7554/elife.105302.1>
- Tyson RA, Epstein DBA, Anderson KI *et al.* High resolution tracking of cell membrane dynamics in moving cells: an electrifying approach. *Math Model Nat Phenom* 2010;**5**:34–55.
- van der Walt S, Schönberger JL, Nunez-Iglesias J *et al.* Scikit-image: image processing in Python. *PeerJ* 2014;**2**:e453.
- van Haastert PJM, Keizer-Gunnink I, Kortholt A. Coupled excitable Ras and F-actin activation mediates spontaneous pseudopod formation and directed cell movement. *Mol Biol Cell* 2017;**28**:922–34.
- Virtanen P, Gommers R, Oliphant TE *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods* 2020;**17**:261–72.
- Xiong Y, Iglesias PA. Tools for analyzing cell shape changes during chemotaxis. *Integr Biol (Camb)* 2010;**2**:561–7.