BMC Bioinformatics

SOFTWARE

Open Access

RABiTPy: an open-source Python software for rapid, AI-powered bacterial tracking and analysis

Samyabrata Sen[1,2,3†], Indraneel Vairagare[1,2,3,4†], Jitendrapuri Gosai[1,2,3] and Abhishek Shrivastava[1,2,3*]

†Samyabrata Sen and Indraneel Vairagare have equal contribution.

*Correspondence: ashrivastava@asu.edu

[1] Center for Fundamental and Applied Microbiomics, Biodesign Institute, Arizona State University, Tempe, AZ 85287, USA
[2] School of Life Sciences, Arizona State University, Tempe, AZ 85287, USA
[3] Center for Biological Physics, Arizona State University, Tempe, AZ 85287, USA
[4] School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ 85287, USA

Abstract

Bacterial tracking is crucial for understanding the mechanisms governing motility, chemotaxis, cell division, biofilm formation, and pathogenesis. Although modern microscopy and computing have enabled the collection of large datasets, many existing tools struggle with big data processing or with accurately detecting, segmenting, and tracking bacteria of various shapes. To address these issues, we developed RABiTPy, an open-source Python software pipeline that integrates traditional and artificial intelligence-based segmentation with tracking tools within a user-friendly framework. RABiTPy runs interactively in Jupyter notebooks and supports numerous image and video formats. Users can select from adaptive, automated thresholding, or AI-based segmentation methods, fine-tuning parameters to fit their needs. The software offers customizable parameters to enhance tracking efficiency, and its streamlined handling of large datasets offers an alternative to existing tracking software by emphasizing usability and modular integration. RABiTPy supports GPU and CPU processing as well as cloud computing. It offers comprehensive spatiotemporal analyses that includes trajectories, motile speeds, mean squared displacement, and turning angles—while providing a variety of visualization options. With its scalable and accessible platform, RABiTPy empowers researchers, even those with limited coding experience, to analyze bacterial physiology and behavior more effectively. By reducing technical barriers, this tool has the potential to accelerate discoveries in microbiology.

Keywords: Bacterial tracking, Bacterial motility, Chemotaxis, Cellular motility, Computational biology, Quantitative biology

Introduction

The accurate and automated tracking of bacterial cells in time-lapse microscopy images and videos is important for gaining mechanistic insights into a wide range of fundamental biological processes. Bacteria exhibit diverse morphologies, movement strategies, and behaviors that underlie key processes such as motility, chemotaxis, cell division, biofilm formation, and pathogenicity [1–4]. Characterizing the spatial and temporal dynamics of individual cells as they navigate complex environments provides crucial insights into how genetic, physiological, environmental, and host-associated factors

© The Author(s) 2025. **Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

drive population-level outcomes. For example, elucidating how bacterial cells respond to gradients of nutrients or antibiotics can inform the design of more effective antimicrobial strategies, while understanding the mechanisms behind coordinated behaviors like swarming or fruiting body formation can uncover new principles in microbial development and ecology.

Over the past decade, a variety of computational approaches have been developed to automate bacterial cell segmentation and tracking by leveraging classical image processing techniques such as global or local thresholding, edge detection, and morphological filtering. Current platforms include, but are not limited to, the TrackMate [5] plugin of ImageJ [6], Oufti [7], YSMR [8], MicrobeTracker [9], and MicrobeJ [10]. Some of the available tracking software rely mostly on traditional segmentation methods that utilize traditional image thresholding, either alone or in combined with fitting methods. However, these techniques can yield inconsistent results, especially when processing cells with non-standard shapes such as filamentous or helical bacterial forms. To address these complexities, recent approaches have begun incorporating algorithms that are pre-trained on diverse cell shapes into the segmentation pipeline. Neural network-based methods, such as those demonstrated by Omnipose [11], trained on microbial images, and CellPose [12], trained on the morphology of various eukaryotic cells, have shown promise in segmenting complex cell shapes and crowded fields of view. These algorithms learn shape priors and texture features directly from annotated training data, often outperforming traditional segmentation methods, especially when dealing with heterogeneous cell populations. Nevertheless, the uptake of these tools can be hindered due to lack of their seamless integration with existing workflows for tracking bacteria in thousands of images generated from a single experiment, a task that is significantly cumbersome for a biologist who has limited training in software development. Furthermore, as experiments increasingly yield massive datasets, either through long-term imaging or parallel acquisition from multiple fields of view, scalability, computational efficiency, and cloud computing have emerged as critical requirements. Recent updates to TrackMate provide additional segmentation plugins that integrate machine learning tools like ilastik [13] and deep learning approaches such as Cellpose [12], StarDist [14] and Omnipose [11] to offer improved segmentation accuracy. While these methods generally work well, on typical personal computers, they tend to slow down considerably when analyzing thousands of densely packed bacterial swarms or images of motile bacteria with low signal-to-noise ratios. Moreover, tools like Weka [15] and ilastik [13] mostly require a training phase for bacterial images. This training process can be time-intensive, particularly for users with limited computational expertise. Even though Omnipose [11] is trained on bacterial images and supports GPU acceleration, and TrackMate has incorporated AI-based segmentation options, users working with large-scale bacterial datasets may still experience performance bottlenecks. For instance, due to the use of CPU memory by the traditional versions of ImageJ [6] and the need for scripting to access TrackMate's batch mode, a non-coder might find it very time-consuming to load thousands of (non-virtual) images from a single bacterial tracking experiment on the software's graphical user interface. In addition, fragmented workflows that involve multiple tools written in different programming languages could lead to compatibility issues in the future while

the requirement of manual steps for data analysis can further complicate the process for biologists lacking extensive computational skills.

In response to these evolving challenges, we introduce RABiTPy (*Rapid Artificially-intelligent Bacterial Image Tracker in Python*), an open-source Python software package designed to simplify bacterial tracking for researchers who may lack extensive coding experience. Unlike many existing solutions, RABiTPy emphasizes a fully integrated pipeline that combines segmentation, tracking, analysis, and visualization within a unified, scriptable interface, streamlining the full workflow and reducing the need to switch between platforms. The package seamlessly integrates a variety of traditional thresholding and AI-based segmentation methods, including Omnipose, with TrackPy [16] (a Python-based tracking library), enabling precise and efficient tracking of bacterial cells from segmented data. By combining the strengths of both traditional and advanced segmentation techniques with robust tracking capabilities within a single, user-friendly platform, RABiTPy provides a consolidated workflow by combining widely used tools into a single pipeline, reducing the need to switch between platforms for an extensive variety of use cases. Whether analyzing sparse populations of bacterial cells or collectively moving monolayers of morphologically diverse species, RABiTPy can handle bacteria exhibiting a range of behaviors, including swimming [17], gliding [18], or twitching [19], either as individual cells [20] or within collective monolayers [21]. Additionally, RABiTPy is able to identify various microbial shapes, ensuring accurate segmentation and tracking across different morphological types.

Beyond its tracking capabilities, RABiTPy allows users to analyze and visualize tracking data in multiple publication-quality formats. It provides a range of data analysis tools that facilitate the extraction of key metrics such as trajectories, motile speeds, mean squared displacement, and turning angles. The software includes a suite of customizable plotting functions that generate high-quality visualizations, including real-time trajectory overlays, spatiotemporal scatter plots, speed distribution histograms, and heatmaps of trajectory density. These features enable researchers to present their findings clearly and effectively, supporting the interpretability and impact of their studies.

By using the Jupyter notebook environment, RABiTPy offers an interactive, accessible, and scriptable interface, lowering the barriers to entry for biologists that wish to track and analyze their favorite microbes. Beyond segmentation and tracking, RABiTPy emphasizes performance and scalability. Its modular architecture enables efficient handling of large datasets and supports both CPU and GPU acceleration along with cloud computing, allowing users to exploit powerful computational resources. Integration with standard scientific Python libraries, such as NumPy [22], Pandas [23], Seaborn [24, 25], scikit-image [26], segmentation library such as Omnipose [11], and the tracking library TrackPy [16] ensures compatibility with a broad ecosystem of analysis tools, while customizable parameter settings and intuitive visualization features support iterative refinement and quality control. Through these capabilities, RABiTPy streamlines the path from raw data to quantitative insights, accommodating a wide variety of experimental systems, imaging modalities, and bacterial species. We demonstrate that RABiTPy accurately segments and tracks multiple bacterial species including but not limited to, *Escherichia coli* [27], *Pseudomonas aeruginosa* [28], *Mycoplasma mobile* [29, 30], *Spiroplasma eriocheiris* [29, 31], *Bacillus subtilis* [32], *Proteus mirabilis* [33], *Flavobacterium*

johnsoniae [34], and *Helicobacter pylori* [4]. These bacteria have different shapes ranging from rods, spherical, filamentous, helical, or irregular and we find that RABiTPy enables precise, large-scale quantification of bacterial dynamics and motility patterns.

RESULTS

Modular workflow of RABiTPy for bacterial tracking

RABiTPy utilizes a robust and modular workflow designed for high-throughput bacterial tracking and analysis (Fig. 1). To enhance user accessibility, it accepts a wide range of input formats, including '.tiff stacks', individual image files, '.avi', and '.mov' (see methods). The software is compatible with time-lapse microscopy images captured using phase-contrast, DIC, or fluorescent microscopes. For segmentation, RABiTPy offers flexibility by allowing users to apply standard thresholding algorithms from the scikit-image [26] library, such as Otsu's method [35], Li's minimum cross-entropy [36], or adaptive local thresholding [37]. These algorithms, documented in the thresholding module of the Python library scikit-image [26], are well-suited for simpler datasets with clear

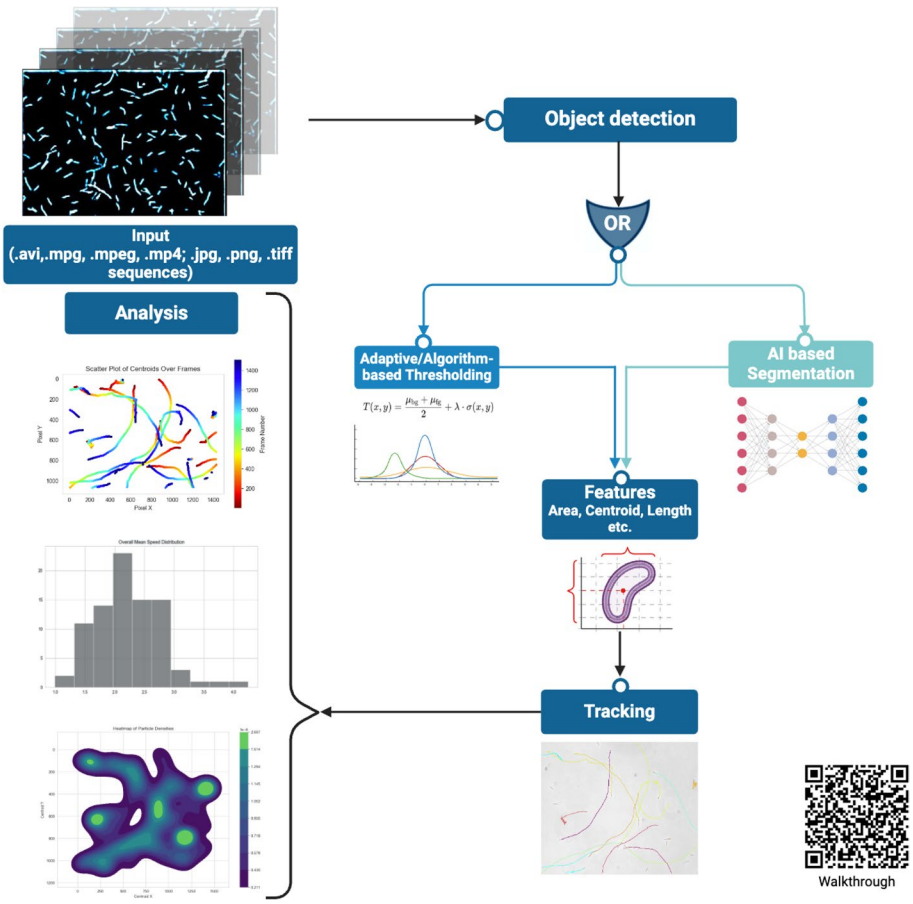

Fig. 1 Overview of the RABiTPy workflow. RABiTPy processes time-lapse microscopy images through segmentation, feature extraction, and motility tracking. The software generates outputs such as trajectory overlays, speed distributions, and trajectory density heatmaps, enabling comprehensive analysis of bacterial motility. The QR code on bottom right links to the walkthrough (example) Jupyter notebook

contrast. For more challenging cases, such as densely populated images or poorly separated objects, RABiTPy integrates Omnipose [11], a neural network-based segmentation algorithm pre-trained on diverse bacterial cell datasets.

By providing two distinct segmentation pathways (Fig. 1), the Jupyter notebook of RABiTPy allows researchers to choose between traditional and deep learning-based segmentation methods depending on their dataset complexity, enabling adaptable workflows across a range of bacterial imaging conditions. Once bacterial cells are detected, the software extracts key features such as area, centroid, and length, forming the foundation for subsequent tracking analyses. RABiTPy delivers a range of visually intuitive outputs, including real-time trajectory overlays on images, spatiotemporal scatter plots of tracks, speed distribution histograms, and heatmaps of trajectory density (Fig. 1). These outputs support both exploratory and quantitative analyses of bacterial motility. RABiTPy builds on existing segmentation algorithms and its strength lies in unifying both traditional and deep learning-based approaches within a single, cohesive, customizable, and user-friendly framework, allowing users to flexibly adapt their workflows to experimental needs and efficiently transition from segmentation to analysis.

RABiTPy is a versatile software for tracking the motility of diverse bacterial species

RABiTPy was tested on a wide range of bacterial species with diverse shapes, sizes, and movement strategies, such as *Mycoplasma mobile* [29, 30], *Spiroplasma eriocheiris* [29, 31], *Proteus mirabilis* [33], *Bacillus subtilis* [32], *Flavobacterium johnsoniae* [34], *Escherichia coli* [27] and *Helicobacter pylori* [4] (Figs. 2 and 3). These species represent a spectrum of bacterial morphologies, from rod-shaped cells to helical forms and motility types, including gliding, swimming, and swarming. RABiTPy accurately tracks the single cell gliding of *M. mobile* and *F. johnsoniae*, the serpentine swimming of *S. eriocheiris*, the complex group dynamics of *B. subtilis* and *P. mirabilis* and the swimming movements of *H. pylori*, and *Escherichia coli.* Swarms of *B. subtilis* posed an extra challenge due to higher density, but when combined with AI-based segmentation, the software performed well while tracking the monolayer of swarming cells (Fig. 3). Quantitative analyses further underscore the performance of RABiTPy across datasets. For example, speed distributions for each species reveal distinct motility ranges, while trajectory density heatmaps and turning angle analyses provide additional layers of insight into bacterial dynamics. By giving the user the option to initiate trajectories from a common origin, RABiTPy ensures consistent comparisons across species, enabling researchers to evaluate motility differences effectively (Figs. 2 and 3).

Detailed tracking of the gliding bacterium *F. johnsoniae* as an example to demonstrate the capability of RABiTPy

We used *F. johnsoniae* UW101, a model organism well-known for its robust gliding motility, as a case study (Fig. 4). Gliding motility, defined by smooth, surface-associated movement interspersed with sudden turns and achieved without the use of flagella or pili, is a hallmark feature of this species [31]. This unique behavior makes *F. johnsoniae* an ideal candidate for evaluating the precision and depth of the tracking capabilities of RABiTPy. Using time-lapse images acquired with a standard phase-contrast microscope, RABiTPy successfully tracked multiple *F. johnsoniae* cells gliding on a glass

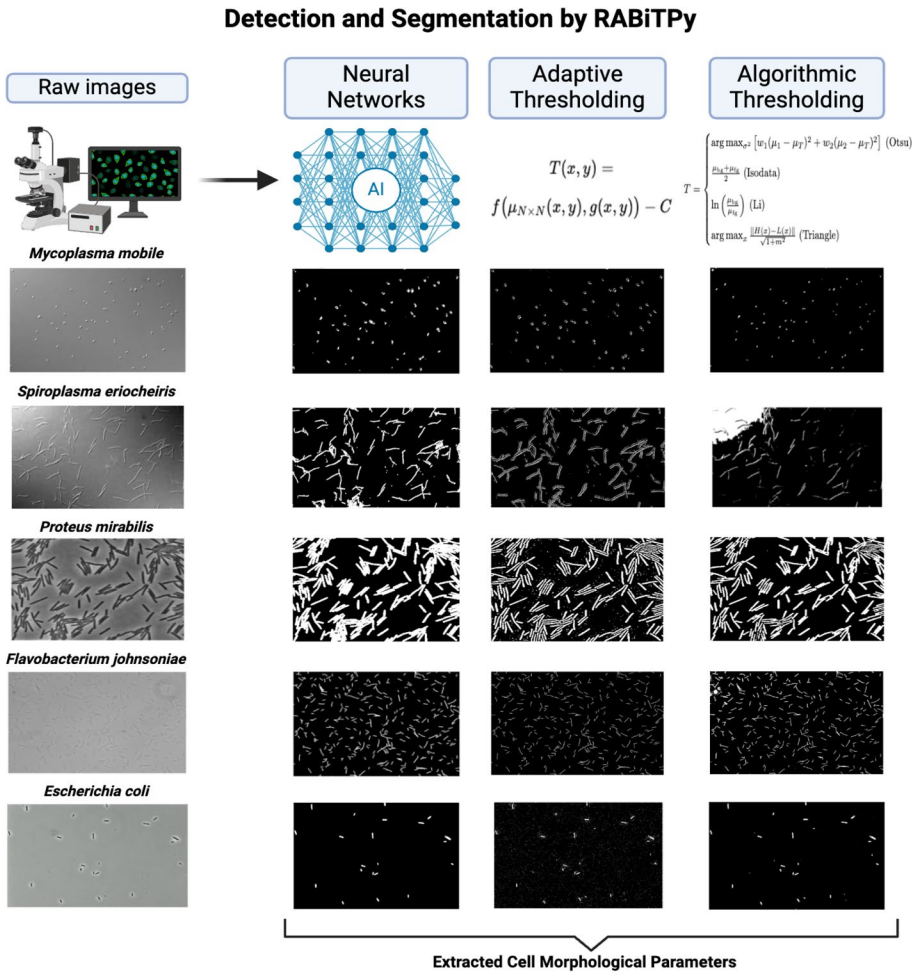

Fig. 2 Usage of traditional thresholding and AI-based methods for segmentation across diverse bacterial species. RABiTPy demonstrates segmentation performance using algorithm-based or adaptive thresholding and AI-based methods on bacterial images with varying shapes and densities. While both approaches enable accurate segmentation, AI-based methods excel in densely populated and crowded environments, ensuring reliable motility tracking

surface generating detailed outputs that quantitatively characterized their motility. For this analysis, the input was a standard AVI format video file and the entire workflow was executed within a Jupyter notebook using RABiTPy's modular pipeline. The video file and Jupyter notebook can be found here: https://github.com/indraneel207/RABiTPy/tree/main/Additional%20examples. Cells were segmented using the integrated Omnipose model with default parameters, followed by morphological filtering to exclude small objects (area < 20 px^2). Tracking was performed using the built-in TrackPy-based linker with max displacement set to 50 pixels and memory set to 50 frames. Trajectories were then filtered to retain only those present for at least 500 frames and exhibiting minimum displacement of 20 pixels. Track smoothing was applied using a Savitzky-Golay [38] filter (window length $= 5$, polyorder $= 2$), and turning angles were computed from simplified trajectories using Ramer–Douglas–Peucker (RDP) [39, 40] with $\varepsilon = 5.0$. Additional analyses were performed on these outputs to highlight the versatility of RABiTPy

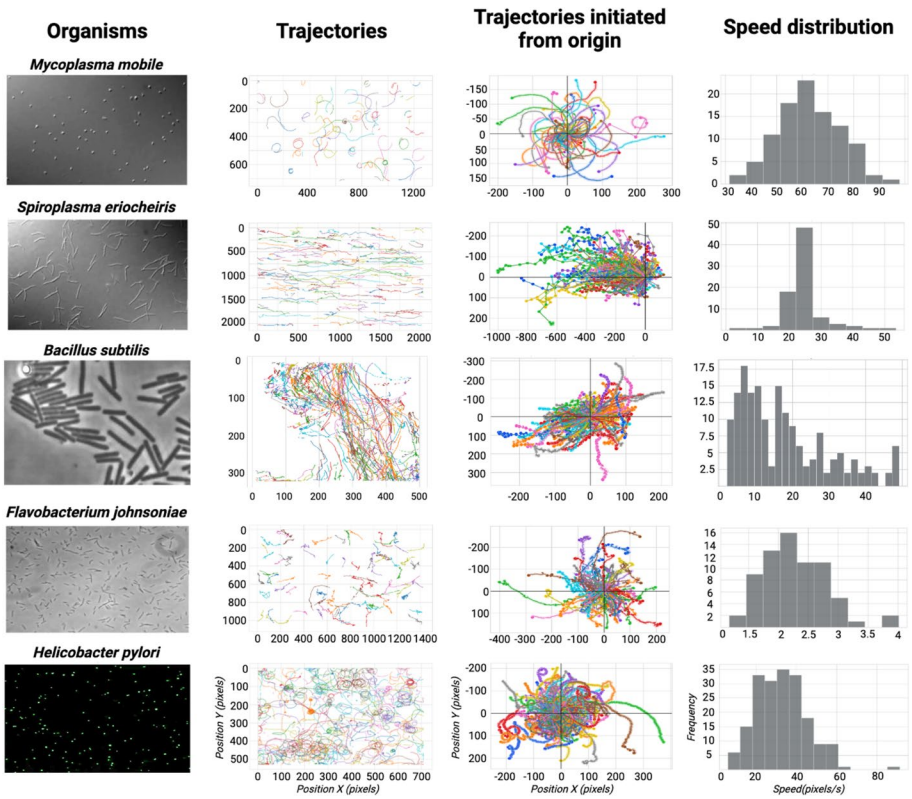

Fig. 3 Tracking the motility of diverse bacterial species. RABiTPy tracks trajectories of bacteria with diverse morphologies, capturing distinct motility behaviors across species. Quantitative outputs such as speed distributions and trajectory density maps demonstrate its ability to analyze motility comprehensively across a wide range of bacterial types

in providing usable data for various subsequent analyses. The time-lapse trajectory overlays (Fig. 4A) clearly distinguish individual movement paths, highlighting the smooth gliding motion of cells across different frames, which also provided a visual validation of tracking fidelity on raw video frames. The mean squared displacement (MSD) analysis (Fig. 4B) revealed consistent motility dynamics, emphasizing the uniformity and robustness of gliding behavior over time. Aspect ratio distributions (Fig. 4C) captured cell morphology variations, highlighting the elongated shapes characteristic of gliding cells and demonstrating how RABiTPy can be used for morphometric filtering or subpopulation analysis. A correlation analysis between cell area and speed (Fig. 4D) revealed a moderate negative relationship ($r = -0.41$, $p = 4.78 \times 10^{-2}$ but there is a wide spread thus suggesting that more data is required before any significant conclusions are made. This trend may be influenced by a combination of physical and mechanical factors, such as altered surface contact or changes in cellular adhesion dynamics related to size [41–43]. Turning angle analysis emphasized the directional persistence of their movement, with cells generally following linear paths and occasional abrupt changes in direction, consistent with their surface-associated motility. Further analysis of the generated trajectory data (Fig. 4E) enabled detailed visualization of both distances traveled and angle changes over time for individual cells, exemplified by a single representative cell. Cell-wise turn

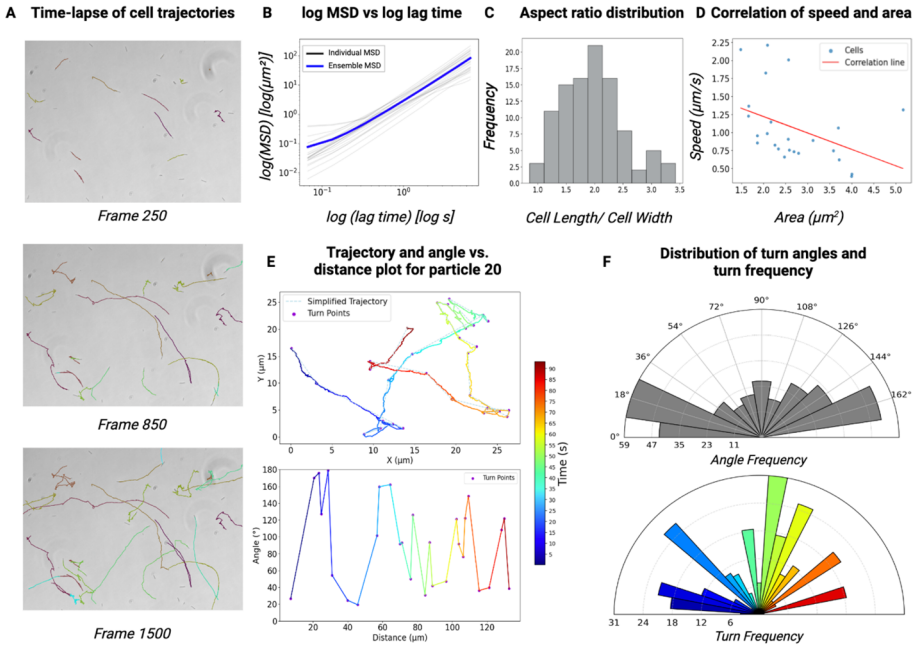

Fig. 4 Detailed analysis of bacterial motility. RABiTPy tracks the motility of the gliding bacterium *F. johnsoniae* as a use case, generating comprehensive quantitative metrics, including (**A**) overlay of RABiTPy trajectories on images at frames 250, 850, and 1500. **B** Log–log plot of mean squared displacement (MSD) versus lag time, showing individual cell trajectories (gray lines) and the ensemble MSD (blue line). **C** Aspect ratio distributions (cell length/cell width) representing the morphology of the cells in the video. **D** Correlation of speed and area, where individual cells are plotted as blue dots, and the red line shows a negative correlation. **E** Trajectory and angle analysis of a representative cell, including the simplified trajectory with turn points (top panel, color-coded by time) and angular changes as a function of distance traveled (bottom panel). **F** Distributions of turn behaviors, with the top panel showing the angle frequency distribution and the bottom panel showing cell-wise turn frequency, where each colored bar represents the turn frequency of a single cell

frequency (Fig. 4F) also provided insights into individual cell behavior, highlighting variability in turning patterns among the population, suggesting behavioral diversity, even within a clonal population, potentially linked to local surface properties or dynamic adhesion regulation [44, 45]. These analyses illustrate how RABiTPy-generated data can support more detailed downstream investigations of bacterial motility. Furthermore, real-time trajectory overlays generated by RABiTPy enhanced the visualization of *F. johnsoniae* movement paths. This end-to-end pipeline demonstrated on a single 1500-frame AVI video (15 fps), can be easily adapted by users for other gliding or motile species with minimal parameter tuning. This feature proved particularly valuable for interpreting dynamic behaviors such as pauses, directional shifts, and clustering, offering a comprehensive understanding of their motility patterns.

Comparisons of RABiTPy-generated trajectories with the ground truth

To test the accuracy of RABiTPy-generated trajectories, three different users manually tracked *Flavobacterium johnsoniae* and *Pseudomonas aeruginosa* [28] cells from two different videos to establish the consensus ground truth. When compared with the consensus ground truth, several key metrics were used to evaluate the tracking performance. Identification F1-Score (IDF1) reflects the harmonic mean of precision

and recall in identity assignment, ensuring accurate tracking over time. Identification Recall (IDR) measures the proportion of correctly tracked identities among the ground truth, indicating the reliability of the tracking algorithm in maintaining object identities. Multiple Object Tracking Accuracy (MOTA) evaluates the overall tracking performance by combining identity switches, missed detections, and false positives, providing a comprehensive measure of tracking accuracy [46]. Mean Intersection over Union (Mean IoU) [47] assesses the spatial alignment between predicted and ground truth segmentation, indicating how well the detected shapes overlap with the actual objects. *F. johnsoniae* tracking achieved IDF1, ID Recall (IDR), MOTA, and Mean IoU values of 0.996, 0.993, 0.993, and 0.994, respectively, reflecting high accuracy in identity management and spatial alignment. Similarly, for *P. aeruginosa*, IDF1, ID Precision (IDP), ID Recall (IDR), and MOTA were 0.978, 0.996, 0.960, and 0.954, with a Mean IoU of 0.994, underscoring precise tracking with minimal errors. It highlights the close alignment between user-tracked and RABiTPy-derived paths, demonstrating minimal deviations (Δd) across multiple cells (Fig. 5). Positional accuracy was further assessed through Euclidean errors and percentage deviations. For *F. johnsoniae*, the overall mean Euclidean error was 0.43 µm (SE: 0.0073 µm), while for *P. aeruginosa,* it was 0.13 µm (SE: 0.0034 µm). Percentage errors showed low deviations for both species—for *F. johnsoniae*, the mean percentage error in the x-coordinate was 0.137% (SE: 0.019%), while the y-coordinate exhibited 0.590% (SE: 0.025%). For *P. aeruginosa*, the mean percentage errors for x and y coordinates were 0.123% (SE: 0.013%)

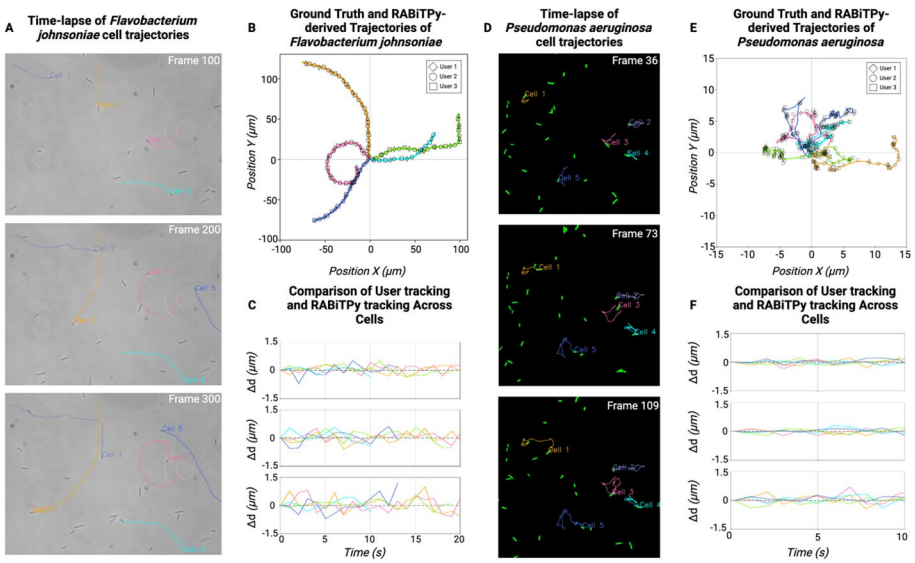

Fig. 5 Validation of the tracking accuracy of RABiTPy. Comparison of user-tracked trajectories and RABiTPy-derived trajectories (colored tracks) demonstrate minimal deviations, highlighting the consistent performance of RABiTPy. **A** Time-lapse images showing trajectories of *F. johnsoniae* cells. **B** Ground truth (represented by diamond, circle, and square markers) and RABiTPy-derived trajectories of *F. johnsoniae*. **C** Positional deviations (Δd) of user-tracked data from RABiTPy trajectories across individual *F. johnsoniae* cells. **D** Time-lapse images showing trajectories of P. aeruginosa cells. **E** Ground truth and RABiTPy-derived trajectories of *P. aeruginosa*. **F** Positional deviations (Δd) of user-tracked data from RABiTPy trajectories across individual *P. aeruginosa* cells

and 0.243% (SE: 0.013%), respectively. These results demonstrate RABiTPy's accuracy and reliability across diverse datasets.

Discussion

The development of RABiTPy addresses longstanding challenges associated with existing tools for bacterial tracking. By adopting a hybrid model that integrates both classical image processing techniques and advanced AI-driven segmentation methods (Fig. 1), RABiTPy provides a robust platform capable of handling diverse and complex datasets that are typical in modern microbiological research. For instance, our results with *P. mirabilis*, *F. johnsoniae*, *H. pylori*, *S. eriocheiris*, *M. mobile*, *E. coli*, *and B. subtilis*, illustrate the ability of RABiTPy to accurately track both individual and collective movements, encompassing swimming, gliding, and swarming behaviors (Figs. 2 and 3). This adaptability ensures that researchers can apply RABiTPy to a wide range of experimental systems without the need for extensive customization or multiple software platforms.

The incorporation of AI-based segmentation methods significantly enhances its performance in challenging imaging conditions. This segmentation accuracy translates directly into more reliable tracking results, thereby enabling precise quantitative analyses of bacterial motility and dynamics. While the robust AI-driven segmentation excels in complex imaging conditions, in future, the performance of pre-trained models can be further optimized for specific experimental setups. To enhance user experience, future versions of RABiTPy could include a broader range of pre-trained models and intuitive tools for custom model training. These enhancements will make advanced segmentation even more accessible, ensuring that all users can achieve optimal results with minimal effort.

One notable advantage of RABiTPy is its scalability and efficiency in processing large datasets. The support for both CPU and GPU processing, coupled with its modular architecture, ensures that it can efficiently manage and analyze large-scale datasets. Additionally, performance limitation due to local computational resources can be easily circumvented through the use of cloud computing platforms such as Google Colab. This capability is essential for high-throughput studies aimed at uncovering complex bacterial behaviors across many conditions and replicates, even on resource-constrained systems. The user-friendly interface of RABiTPy, facilitated by its operation within Jupyter notebooks, democratizes access to advanced bacterial tracking capabilities. By lowering the barrier to entry for researchers without extensive programming expertise, RABiTPy enables a broader segment of the microbiology community to leverage sophisticated tracking and analysis tools. The interactive nature of Jupyter notebooks allows users to iteratively refine their analysis workflows, visualize intermediate results, and customize parameters in real-time, fostering a more intuitive and efficient research process. Additionally, the provision of publication-ready visualization options ensures that researchers can seamlessly transition from data analysis to manuscript preparation, enhancing the overall workflow efficiency. Overall, RABiTPy provides an integrated pipeline for bacterial segmentation, tracking, visualization, and analysis. Designed for users without extensive programming experience, it facilitates exploration of fundamental microbiological questions. Its capacity to handle large, diverse datasets supports high-throughput

screening and systems biology approaches, accelerating discoveries in microbial physiology and ecology. As the tool evolves, it is well positioned to support deeper investigations into microbial life and its interactions with the environment and host organisms.

Methods

Software, hardware, and installation guidelines

RABiTPy is compatible with Windows, macOS, and Linux, and it requires Python 3.10.11. The tool is typically run within Jupyter Notebooks. A detailed Jupyter Notebook is available as a walkthrough in our GitHub https://github.com/indraneel207/RABiTPy/blob/main/walkthrough.ipynb

and a PDF file of the walkthrough is available as supplementary text.

This walkthrough ensures that the user can install RABiTPy and its required Python libraries using pip and runs example files. The recommended installation command for use within a Jupyter notebook is also provided in our GitHub README: https://github.com/indraneel207/RABiTPy. Also, RABiTPy is available on PyPI (https://pypi.org/project/RABiTPy/) and a PDF of the PYPI page is available as supplementary text—users can install it directly in Jupyter notebook or other Python environment using the command: "pip install RABiTPy". Before proceeding with their own tasks, it is recommended that the user run through the walkthrough and the included example file to ensure that their environment is set up correctly.

Supported datatypes

RABiTPy enables seamless loading and processing of both video files and image sequences, ensuring compatibility with a wide range of microscopic datasets. It converts frames extracted from microscopic movies or image sequences into NumPy arrays [23], making them ready for analysis (Figure S1). The tool supports common video formats—AVI, MP4, MPG, and MPEG thus offering flexibility for various experimental setups. By default, RABiTPy operates at 15 frames per second and uses a pixel scale factor of 1, but users can interactively adjust these settings to meet their specific needs. For experiments that produce individual images rather than continuous video, RABiTPy also supports importing PNG, JPG, and TIFF files. To ensure reproducibility and accurate quantification, we recommend that users retain the original acquisition resolution, regardless of the file format used.

Segmentation and detection of microbial cells

RABiTPy offers a precise framework for cell detection within image frames, providing flexibility through two distinct methods: thresholding and AI-based detection using Omnipose [11], an algorithm that is pre-trained with images of diverse microbes. Thresholding options include manual, adaptive, and algorithm-driven methods, enabling microbiology researchers to select the approach best aligned with their experimental conditions and objectives (Figure S2). The manual thresholding method is typically recommended when the users are more familiar with their datasets and can specify a fixed threshold value to classify pixels into foreground and background, making it particularly useful for datasets with consistent lighting and contrast.

Algorithm-based thresholding dynamically determines the ideal threshold by analyzing the intrinsic statistical or geometric attributes of an image. RABiTPy equips users with a broad selection of such techniques, enabling precise customization to meet specific analytical demands. These methods are outlined below: *(1) Otsu's Method* [35]. It is particularly suited for tasks that require maximizing the between-class variance (σ^2) to distinguish foreground from background elements in an image. This approach can be defined by the following equation: $\sigma^2 = w_1(\mu_1 - \mu_T)^2 + w_2(\mu_2 - \mu_T)^2$. Here, w_1 and w_2 represent the probabilities of the two classes (background and foreground), μ_1 and μ_2 denote the mean intensities of the background and foreground pixels, respectively, and μ_T signifies the total mean intensity of the image. *(2) Isodata* [48]*:* It iteratively calculates the threshold (T) as the midpoint of the mean intensities of the background (μ_{bg}) and foreground (μ_{fg}): $T = (\mu_{bg} + \mu_{fg})/2$.*(3) Li's method* [36]*:* It utilizes the logarithmic ratio of mean intensities to determine the threshold, where (T) represents the threshold, μ_{bg} is the mean intensity of the background pixels, and μ_{fg} is the mean intensity of the foreground pixels: $T = \ln(\mu_{bg}/\mu_{fg})$. *(4) Triangle Method* [26] identifies the threshold T as the histogram bin where the distance $D(x)$ between the histogram and a line connecting the peak and endpoint of the histogram is greatest. $D(x) = \| H(x) - L(x) \| \Big/ \sqrt{1 + m^2}$ where x represents the bin index in the histogram. $H(x)$ denotes the value of the histogram at bin x. $L(x)$ represents the value of the line connecting the peak and endpoint of the histogram. m is the slope of the line $L(x)$ and $D(x)$ specifies the distance from the histogram $H(x)$ to the line $L(x)$. The threshold T is then identified as the bin where the distance $D(x)$ is the largest.

Adaptive thresholding [37] computes thresholds for different regions of the image, making it effective for scenarios with non-uniform illumination by ensuring localized adaptability. Gaussian adaptive thresholding, for instance, computes the threshold $T(x, y)$ for each pixel based on the mean intensity. $T(x, y) = \mu_{N\times N}(x, y) - C$. Here, $\mu_{N\times N}$ is the mean intensity of the $N \times N$ neighborhood centered at (x, y) and C is a a constant subtracted from the mean to determine the threshold.

RABiTPy also integrates the multiple pre-trained models of Omnipose [11]. For example, the *bact phase omni* model is tailored for bacterial segmentation in phase-contrast images, while *bact fluor omni* is designed for bacterial segmentation in fluorescence microscopy. Additionally, the *bact phase omni 2* model provides a simplified solution for single-channel bacterial phase-contrast imaging. All these deep neural network models are integrated into RABiTPy and can be applied directly within the same Jupyter notebook.

Feature extraction

After detecting and segmenting the cells using the above-described methods, RABiTPy can extract a comprehensive set of features for detailed object analysis. These include spatial properties (e.g., area, bounding box, centroid, long axis, short axis), shape descriptors (e.g., eccentricity, equivalent diameter, perimeter), and intensity-based measures (e.g., mean, minimum, maximum intensity, weighted centroid). The centroid $(C_x, C_y) = \sum_i (x_i, y_i) I(x_i, y_i) \Big/ \sum_i I(x_i, y_i)$ is the weighted mean position of pixels within a segmented region. Its calculation provides a precise geometric center for each cell,

which RABiTPy uses to determine cell trajectories across frames later. Here, (C_x, C_y) denotes the centroid coordinates, (x_i, y_i) are the coordinates of each pixel i in the region, and $I(x_i, y_i)$ is the intensity of pixel$i$. To further improve centroid accuracy, an option has been added to fit a 2D Gaussian to the detected features. This refinement is particularly beneficial for low-magnification videos where bacteria appear more circular, enhancing localization precision [49]. The 2D Gaussian function used for refinement is defined as:

$$G(x,y) = A\exp\left(-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)\right) + C$$

where A is the amplitude, (x_0,y_0) is the centroid, σ_x,σ_y are the standard deviations in x and y, and C is the offset.

Tracking of microbial cells

The DataFrame generated after feature extraction of microbial cells of diverse morphologies is fed into the algorithm of the Python package TrackPy [17] to accurately track particle positions across successive frames. The algorithm identifies the closest matching particle in the next frame based on spatial proximity, minimizing the likelihood of mismatches. Proximity is assessed using highly configurable parameters set by the user, including the maximum allowable distance between consecutive positions, tolerance for missing frames, and centroid coordinates as reference points for linking particles. These parameters ensure reliable trajectory construction, even in densely populated systems. The resulting trajectories are organized into a structured format, associating particle identifiers with their movements and enabling users to derive meaningful insights into dynamic motion patterns. Advanced filtering options refine the analysis, allowing users to exclude particles with minimal movement or insufficient presence across frames. This ensures that the dataset remains focused on significant and actionable motion data (Figure S3).

Visualization and analysis of tracking data

In addition to its versatile microbial detection and tracking capabilities, RABiTPy enhances the user experience with a powerful video overlay feature. This allows the user to visualize particle trajectories, where the labels correspond to particle IDs from the generated CSV, directly on image or video frames, uniquely coloring particles and displaying cumulative tracks over time. By seamlessly combining visualization with analysis, users gain an intuitive and dynamic understanding of particle motion across the video, enabling real-time validation of tracking results.

RABiTPy package provides a comprehensive framework for analyzing cell speed distributions. It is specifically designed to process motion data, offering tools for calculating mean speeds, fitting statistical distributions to speed data, visualizing speed patterns, and exporting analyzed results for further use. By default, it employs a normal distribution for data fitting, but it is versatile enough to support alternative distributions as needed. The visualization is structured as a grid of plots, with customizable parameters to adjust the number of plots displayed per row and the type of statistical model used for

fitting speed data. Default settings include four plots per row and a normal distribution for fitting – with options for alternative curve fittings provided to the user. Additionally, the range of x-axis (swimming speed) values for Gaussian fitting is customizable, with the default implementation automatically selecting the 5% and 95% confidence interval (CI) limits if no user-defined range is provided. The computed mean speeds are stored in a structured format, allowing further analysis or integration into comparative studies (Figure S4). This analysis provides insights into the general dynamics of the system, revealing trends and patterns that emerge from the collective motion. The walkthrough file includes additional notebooks for such applications. For example, we provide the codes to calculate Mean Squared Displacement (MSD) values from the particle trajectories data frame generated by RABiTPy.

(https://github.com/indraneel207/RABiTPy/blob/main/Additional%20examples/Combining%20Multiple%20Tracks.ipynb).

This involves determining the displacement for each time lag and computing ensemble averages. These values can then be visualized using a log–log plot of MSD versus time, providing insights into the diffusion behavior of the cells. Additionally, turn-points can be identified by smoothing trajectories with filters such as the Savitzky-Golay [38] or Ramer-Douglas-Peucker algorithms [39, 40] from SciPy [50], followed by detecting angular changes above a defined threshold.

(https://github.com/indraneel207/RABiTPy/tree/main/Additional%20examples).

The features of these turn-points are illustrated in Fig. 4E and F. These integrated analyses offer a comprehensive understanding of motility patterns and the dynamic behavior of cells within the dataset. Therefore, the breadth of outputs generated by RABiTPy facilitates the study of bacterial motility and offers a single platform for researchers to perform analyses tailored to specific questions, such as turn behaviors, speed correlations, and morphological dependencies. Its adaptability to different motility types further highlights its potential in studying a wide range of motile organisms.

Tracking accuracy evaluation

Tracking accuracy was evaluated using representative datasets of *Flavobacterium johnsoniae* and *Pseudomonas aeruginosa* [29]. For each species, five individual cell tracks were selected per video. Three independent users manually tracked each of the five cells in each video using ImageJ. To generate a consensus ground truth, the x–y positions recorded by the users were averaged frame-by-frame for each cell. This approach minimized user bias and established a robust reference trajectory for comparison. RABiTPy was then used to segment and track the same cells using Omnipose for segmentation and TrackPy for trajectory generation. Using several standard performance metrics, the resulting trajectories were compared against the consensus ground truth. Identification F1 Score (IDF1) quantified the harmonic mean of precision and recall in identity assignment. Identification Recall (IDR) and Identification Precision (IDP) assessed how accurately cell identities were maintained across frames. Multiple Object Tracking Accuracy (MOTA) measured overall tracking accuracy by accounting for missed detections, false positives, and identity switches. Mean Intersection over Union (Mean IoU) evaluated

the spatial overlap between segmented masks and the manually annotated ground truth. Finally, positional accuracy was assessed using Euclidean error and percentage deviations in both x and y coordinates.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-025-06145-w.

Additional file 1

Acknowledgements
Figures were created using biorender.com. RABiTPy stands on the shoulders of Python libraries such as NumPy, Matplotlib, Seaborn, Pandas, Omnipose, and TrackPy – we thank the developers of these libraries for their awesomeness.

Author contributions
SS, IV, JG, and AS conceptualized the project. SS and AS wrote most of the manuscript, with significant inputs from JG and IV. AS, JG, and SS developed preliminary versions of the tracking software. IV then created the RABiTPy software and incorporated multiple modalities, guided by feedback from SS, JG, and AS.

Funding
AS was supported by NIH-NIGMS MIRA award R35GM147131.

Availability of data and materials
No datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Availability and requirements
Project name: RABiTPy: Project home page: https://pypi.org/project/RABiTPy/: Operating system(s): Platform independent: Programming language: Python: Other requirements: Python less that 3.11, greater than or equal to 3.10: License: MIT license: Any restrictions to use by non-academics: license needed.

Competing interests
The authors declare no competing interests.

Received: 31 January 2025 Accepted: 17 April 2025
Published online: 18 May 2025

References

1. Colin R, Ni B, Laganenka L, Sourjik V. Multiple functions of flagellar motility and chemotaxis in bacterial physiology. FEMS Microbiol Rev. 2021;45:038.
2. Taktikos J, Stark H, Zaburdaev V. How the motility pattern of bacteria affects their dispersal and chemotaxis. PLoS ONE. 2013;8: e81936.
3. Ciofu O, Moser C, Jensen PØ, Høiby N. Tolerance and resistance of microbial biofilms. Nat Rev Microbiol. 2022;20:621–35.
4. Aihara E, et al. Motility and chemotaxis mediate the preferential colonization of gastric injury sites by helicobacter pylori. PLoS Pathog. 2014;10: e1004275.
5. Tinevez J-Y, et al. TrackMate: an open and extensible platform for single-particle tracking. Methods. 2017;115:80–90.
6. Schindelin J, et al. Fiji: an open-source platform for biological-image analysis. Nat Methods. 2012;9:676–82.
7. Paintdakhi A, et al. Oufti: an integrated software package for high-accuracy, high-throughput quantitative microscopy analysis. Mol Microbiol. 2016;99:767–77.
8. Schwanbeck J, et al. YSMR: a video tracking and analysis program for bacterial motility. BMC Bioinformatics. 2020;21:166.
9. Sliusarenko O, Heinritz J, Emonet T, Jacobs-Wagner C. High-throughput, subpixel precision analysis of bacterial morphogenesis and intracellular spatio-temporal dynamics. Mol Microbiol. 2011;80:612–27.
10. Ducret A, Quardokus EM, Brun YV. MicrobeJ, a tool for high throughput bacterial cell detection and quantitative analysis. Nat Microbiol. 2016;1:1–7.
11. Cutler KJ, et al. Omnipose: a high-precision morphology-independent solution for bacterial cell segmentation. Nat Methods. 2022;19:1438–48.
12. Stringer C, Wang T, Michaelos M, Pachitariu M. Cellpose: a generalist algorithm for cellular segmentation. Nat Methods. 2021;18:100–6.

13. Berg S, et al. ilastik: interactive machine learning for (bio)image analysis. Nat Methods. 2019;16:1226–32.
14. Schmidt U, Weigert M, Broaddus C, Myers G. Cell Detection with Star-Convex Polygons. Cham: Springer International Publishing; 2018.
15. Witten IH, Frank E, Hall MA, Pal CJ. Data Mining, Fourth Edition: Practical Machine Learning Tools and Techniques. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 2016.
16. Allan, D. B., Caswell, T., Keim, N. C., van der Wel, C. M. & Verweij, R. W. soft-matter/trackpy: v0.6.4. Zenodo https://doi.org/10.5281/ZENODO.12708864 (2024).
17. Wadhwa N, Berg HC. Bacterial motility: machinery and mechanisms. Nat Rev Microbiol. 2022;20:161–73.
18. Shrivastava A, Berg HC. A molecular rack and pinion actuates a cell-surface adhesin and enables bacterial gliding motility. Sci Adv. 2020;6:6616.
19. Corral J, et al. Importance of twitching and surface-associated motility in the virulence of Acinetobacter baumannii. Virulence. 2021;12:2201–13.
20. Davidson CJ, Surette MG. Individuality in Bacteria. Annu Rev Genet. 2008;42:253–68.
21. Mattingly HH, Emonet T. Collective behavior and nongenetic inheritance allow bacterial populations to adapt to changing environments. Proc Natl Acad Sci. 2022;119: e2117377119.
22. Harris CR, et al. Array programming with NumPy. Nature. 2020;585:357–62.
23. Team, T. pandas development. pandas-dev/pandas: Pandas. Zenodo https://doi.org/10.5281/zenodo.13819579 (2024).
24. Waskom, M. *et al.* mwaskom/seaborn: v0.8.1 (September 2017). Zenodo https://doi.org/10.5281/ZENODO.883859 (2017).
25. Hunter JD. Matplotlib: A 2D graphics environment. Comput Sci Eng. 2007;9:90–5.
26. Van Der Walt S, et al. scikit-image: image processing in Python. PeerJ. 2014;2: e453.
27. Mattingly HH, Kamino K, Machta BB, Emonet T. Escherichia coli chemotaxis is information limited. Nat Phys. 2021;17:1426–31.
28. Madukoma CS, et al. Single cells exhibit differing behavioral phases during early stages of pseudomonas aeruginosa swarming. J Bacteriol. 2019. https://doi.org/10.1128/jb.00184-19.
29. Miyata, M. Miyata lab. http://www.sci.osaka-cu.ac.jp/~miyata/index.html.
30. Mizutani M, Tulum I, Kinosita Y, Nishizaka T, Miyata M. Detailed analyses of stall force generation in mycoplasma mobile gliding. Biophys J. 2018;114:1411–9.
31. Sasajima Y, Miyata M. Prospects for the mechanism of spiroplasma swimming. Front Microbiol. 2021. https://doi.org/10.3389/fmicb.2021.706426.
32. Jose A, et al. Immobility of isolated swarmer cells due to local liquid depletion. Commun Phys. 2025;8:1–11.
33. Little K, Austerman J, Zheng J, Gibbs KA. Cell shape and population migration are distinct steps of *Proteus mirabilis* swarming that are decoupled on high-percentage agar. J Bacteriol. 2019. https://doi.org/10.1128/JB.00726-18.
34. Shrivastava A, Johnston JJ, van Baaren JM, McBride MJ. Flavobacterium johnsoniae GldK, GldL, GldM, and SprA are required for secretion of the cell surface gliding motility Adhesins SprB and RemA. J Bacteriol. 2013;195:3201–12.
35. Otsu N. A threshold selection method from Gray-level histograms. IEEE Trans Syst Man Cybern. 1979;9:62–6.
36. Li CH, Tam PKS. An iterative algorithm for minimum cross entropy thresholding. Pattern Recognit Lett. 1998;19:771–6.
37. Bradley D, Roth G. Adaptive thresholding using the integral image. J Graph Tools. 2007;12:13–21.
38. Savitzky A, Golay MJE. Smoothing and differentiation of data by simplified least squares procedures. Anal Chem. 1964;36:1627–39.
39. Ramer U. An iterative procedure for the polygonal approximation of plane curves. Comput Graph Image Process. 1972;1:244–56.
40. Douglas DH, Peucker TK. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica. 1973;10:112–22.
41. Nan B, McBride MJ, Chen J, Zusman DR, Oster G. Bacteria that glide with helical tracks. Curr Biol. 2014;24:R169–73.
42. Persat A, et al. The mechanical world of bacteria. Cell. 2015;161:988–97.
43. Martinez VA, et al. Flagellated bacterial motility in polymer solutions. Proc Natl Acad Sci. 2014;111:17771–6.
44. Mignot T, Shaevitz JW, Hartzell PL, Zusman DR. Evidence that focal adhesion complexes power bacterial gliding motility. Science. 2007;315:853–6.
45. Faure LM, et al. The mechanism of force transmission at bacterial focal adhesion complexes. Nature. 2016;539:530–5.
46. Bernardin K, Stiefelhagen R. Evaluating multiple object tracking performance: the CLEAR MOT metrics. EURASIP J Image Video Proc. 2008;2008:1–10.
47. Rezatofighi, H. *et al.* Generalized intersection over union: a metric and a loss for bounding box regression. In 2019. https://doi.org/10.48550/arXiv.1902.09630.
48. Ridler TW, Calvard S. Picture thresholding using an iterative selection method. IEEE Trans Syst Man Cybern. 1978;8:630–2.
49. Thompson RE, Larson DR, Webb WW. Precise nanometer localization analysis for individual fluorescent probes. Biophys J. 2002;82:2775–83.
50. Virtanen P, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods. 2020;17:261–72.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.