Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 23(11): 3241–3272.
DOI: 10.3934/mbe.2026112
Received: 12 July 2026
Revised: 02 September 2026
Accepted: 21 September 2026
Published: 03 November 2026

Research article

Change-in-velocity detection in multidimensional data

Linh Do[1,*], Dat Do[2], Keisha J. Cook[3], Yitao Shen[4] and Scott A. McKinley[5]

[1] Department of Biostatistics and Bioinformatics, Duke University School of Medicine, 2424 Erwin Road, Durham, NC 27705, USA

[2] Department of Statistics, University of Chicago, 5747 S. Ellis Avenue, Chicago, IL 60637, USA

[3] School of Mathematical and Statistical Sciences, Clemson University, O-110 Martin Hall, 220 Parkway Drive, Clemson, SC 29634, USA

[4] Biochemistry and Molecular Biology and the Huck Institutes of the Life Sciences, The Pennsylvania State University, University Park, PA 16802, USA

[5] Department of Mathematics, Tulane University, 6823 St. Charles Avenue, New Orleans, LA 70118, USA

* **Correspondence:** Email: thuylinh.do@duke.edu; Tel: +1-504-493-4897.

Abstract: In this work, we introduce continuous piecewise-linear approximation via stochastic search (CPLASS), an algorithm for detecting changes in velocity within multidimensional data. The one-dimensional version of this problem is known as the change-in-slope problem. Unlike traditional changepoint detection methods that focus on changes in mean, detecting changes in velocity requires a specialized approach due to continuity constraints and parameter dependencies, which frustrate popular algorithms like binary segmentation and dynamic programming. To overcome these difficulties, we introduce a tailored penalty function to balance improvements in likelihood due to model complexity and a Markov chain Monte Carlo (MCMC)-based approach with tailored proposal mechanisms for efficient parameter exploration. Our method is particularly suited for analyzing intracellular transport data, where the multidimensional trajectories of microscale cargo are driven by teams of molecular motors that undergo complex biophysical transitions. To ensure biophysical realism in the results, we introduce a speed penalty that discourages overfitting of short noisy segments while maintaining consistency in the large-sample limit. Additionally, we introduce a summary statistic called the cumulative speed allocation (CSA), which is robust with respect to idiosyncrasies of changepoint detection while maintaining the ability to discriminate between biophysically distinct populations.

Keywords: changepoint detection; intracellular transport; Markov chain Monte Carlo (MCMC); multidimensional time series; penalty functions; trajectory analysis

1. Introduction

The intracellular transport of organelles and cargo is a fundamental requirement of cellular function. Molecular motors carry lysosomes, vesicles, and other microscale cargo along microtubule filaments, producing trajectories marked by periods of directed motion, thermal fluctuation, and near-stationary pausing [1–3]. Understanding how transport properties change over time, in particular, detecting when a particle transitions between distinct velocity states, is central to characterizing the biophysical behavior of motor proteins and their collective dynamics. These transitions manifest as changes in the velocity of particle position over time, making this a multidimensional generalization of what is known in statistics as the change-in-slope problem.

Changepoint detection has been studied for over sixty years across many scientific domains, including signal processing [4], financial analysis [5, 6], genomics [7, 8], and environmental science [9]. The canonical formulation is detecting changes in the mean of independent observations, a problem for which many effective methods exist. However, the change-in-velocity problem poses a distinct set of challenges. Because particle positions are continuous in time, the increment process cannot be treated as simply changing in mean: Continuity constraints couple parameters across segments in ways that frustrate the most widely used algorithms. Binary segmentation, for instance, iteratively searches for a single changepoint at a time; in a velocity-change setting, an initial estimate placed between two true changepoints cannot be corrected in subsequent iterations, leading to spurious detections (Figure 2). Dynamic programming approaches such as pruned exact linear time (PELT) [10] and optimal partitioning [11] are similarly unsuitable because the continuity requirement introduces parameter dependencies that violate the independence assumptions these methods rely on [12]. Taking first differences to convert velocity changes into mean changes is another natural idea, but [13] showed this approach discards information and performs poorly in practice.

Existing methods for the change-in-slope problem include trend filtering [14], the narrowest-over-threshold (NOT) algorithm [15], continuous-piecewise-linear pruned optimal partitioning (CPOP) [12], and narrowest significance pursuit (NSP) [16]. These methods were developed primarily for scalar time series, although the general NOT framework can be extended to other settings. Coordinate-wise application to multidimensional particle trajectories is possible, but it produces separate changepoint sets that must subsequently be reconciled rather than a single segmentation estimated jointly across coordinates. This distinction is important for intracellular transport trajectories, which are inherently two- or three-dimensional.

In studies of intracellular transport, trajectory-level summaries such as mean squared displacement (MSD) are widely used [17, 18], but they average over entire trajectories and can miss short-lived state transitions. More recent single-particle methods therefore seek to localize changes in diffusivity [19, 20], velocity [21, 22], or both simultaneously [23]. Among related slope-segmentation approaches, piecewise linear approximation of noisy trajectories (PLANT) [24] reconstructs scalar noisy traces as adjacent linear segments and has been applied to coordinate traces obtained from cell-tracking experiments. PLANT does not constrain neighboring fitted segments to meet continuously and can therefore represent an instantaneous positional jump at a detected changepoint. In a distinct drift–diffusion framework, [23] used likelihood-ratio tests and recursive binary segmentation to detect changes in velocity, diffusion, or both, and extended their procedure to multidimensional trajectories. Their method operates on displacement increments rather than estimating a globally continuous piecewise-

linear latent-position trajectory. Thus, continuity of a fitted latent path, joint segmentation across coordinates, and simultaneous selection of the complete changepoint configuration remain distinct modeling and computational considerations.

In previous work, our group [22, 25] used the Bayesian changepoint (`bcp`) algorithm to segment particle paths, modeling the segments as discontinuous piecewise-linear functions plus stationary noise. Although effective for many purposes, this discontinuous model can miss short, fast motile segments that are biophysically meaningful (Figure 3). Enforcing continuity of the latent trajectory is therefore not merely a mathematical distinction: It reflects the physical requirement that cargo position evolves continuously in time and avoids an important blind spot of the discontinuous approximation.

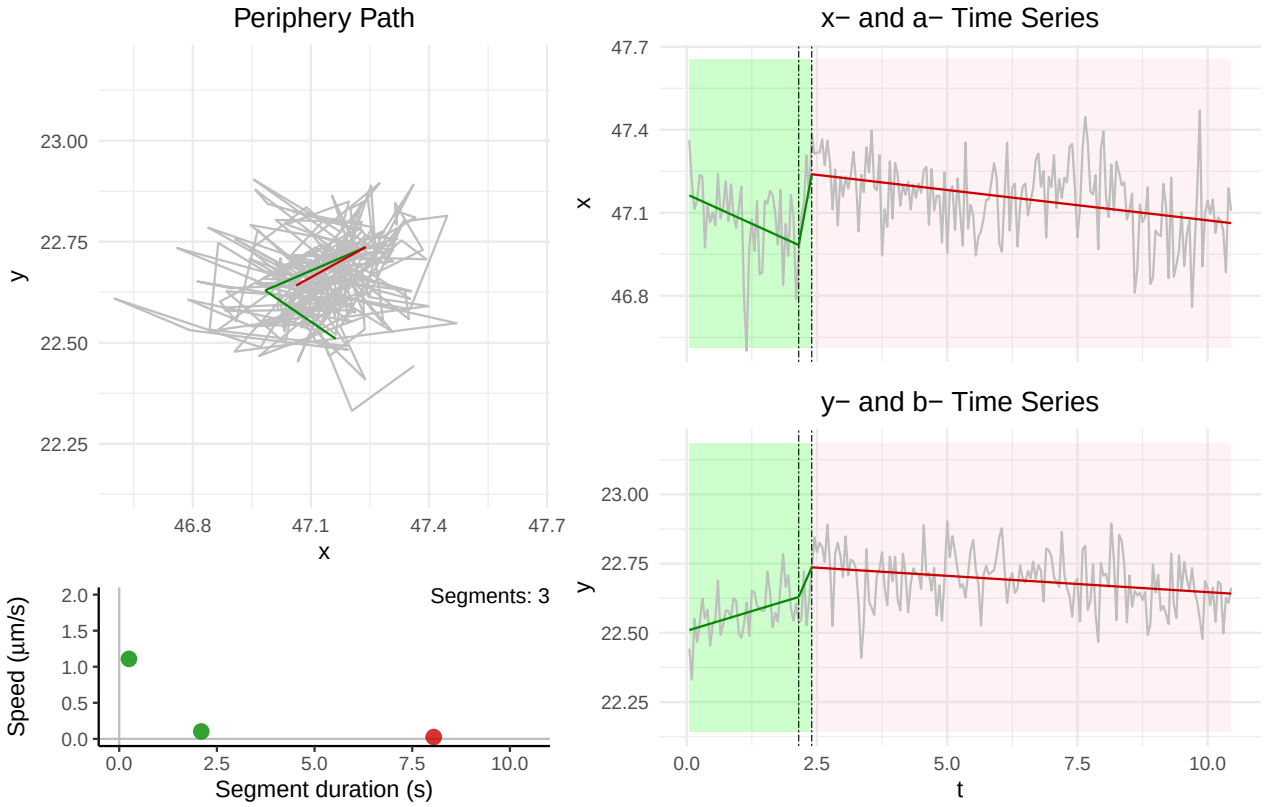


Figure 1. *Representative lysosome trajectory in the periphery of a cell from [25].* The gray curves show the observed trajectory in the x–y plane and the corresponding coordinate-wise time series. The green and red curves denote the inferred anchor positions, colored by inferred state: green indicates motile segments, and red indicates stationary segments. Green and red shaded regions indicate motile and stationary segments, respectively. In the segment-duration versus speed plot, point colors correspond to the inferred segment state.

To address these challenges, we introduce continuous piecewise-linear approximation via stochastic search (CPLASS), an algorithm for detecting changes in velocity in multidimensional particle trajectories. CPLASS models observed positions as independent Gaussian fluctuations around a continuous piecewise-linear anchor trajectory and uses Markov chain Monte Carlo (MCMC) to search over the space of changepoint configurations for the segmentation that optimizes a penalized model-selection

criterion. The criterion combines a strengthened Schwarz information criterion (sSIC) penalty [15, 26] with an optional dimensionless speed penalty that discourages segments whose fitted speeds exceed a physically specified reference speed. We also introduce specialized proposal mechanisms, including a paired changepoint proposal, that are tailored to the structure of the change-in-velocity problem and substantially improve convergence when short, fast segments are present. To reduce sensitivity to stochastic initialization, the implementation uses multiple deterministic starts and retains the highest-scoring solution.

Figure 1 illustrates the CPLASS output for one experimental lysosome trajectory from the periphery of a cell. This example shows how CPLASS detects changes in velocity by estimating a continuous anchor path and partitioning the observed trajectory into distinct segments. For visualization, we label segments with inferred speed greater than 100 nm/s as motile and all remaining segments as stationary; this threshold is not used by the segmentation algorithm itself. The figure provides a concrete example of the segmentation structure produced by CPLASS before we systematically evaluate the method on simulated and experimental data.

We demonstrate the effectiveness of CPLASS on simulated trajectories and three experimental datasets: lysosomal transport in epithelial cells [25]; quantum dot transport driven by kinesin-1, dynein-dynactin-BicD2 (DDB), and kinesin-1/DDB motor pairs [22]; and a preliminary application to end-binding protein 1–green fluorescent protein (EB1-GFP) microtubule endpoint trajectories in neurons [27, 28]. To summarize segmentation output in a way that is robust to the idiosyncrasies of changepoint detection, we use the cumulative speed allocation (CSA) statistic [29], which quantifies the proportion of time a particle spends at each speed and enables principled comparison across experimental groups without requiring an arbitrary motility threshold.

Our contributions are threefold. First, we introduce CPLASS, a continuity-preserving method for detecting velocity changepoints in multidimensional intracellular trajectories. Second, we incorporate a biologically motivated speed penalty to discourage segments that require implausible object velocities while still preserving short active transport events. Third, we demonstrate the utility of the method through simulated trajectories and experimental datasets, and summarize inferred transport behavior using the CSA statistic.

Datasets. We used the following datasets to validate the proposed method. Throughout the paper, when we state that trajectories are observed or simulated at a given frequency (e.g., 20 or 25 Hz), this means that the time step between consecutive observations is the inverse of the sampling frequency (e.g., 0.05 or 0.04 s, respectively). The data include: (1) BS-C-1 monkey kidney epithelial cells and A549 human lung epithelial cells [25] (obtained from the Duke University Cell Culture Facility). For these datasets, intracellular transport is a requirement of cellular functions related to lysosomes; (2) datasets featuring quantum dots being transported by a single kinesin-1 motor, a single dynein-dynactin-BicD2 (DDB) motor, and a kin1-DDB pair [22], involving microtubule-based molecular motor transport; (3) MTEB1 microtubule endpoint tracking data, derived from live-cell imaging of neurites in studies of microtubule polarity [27, 28], consisting of two-dimensional particle trajectories from EB1-GFP comet tracking; and (4) additional simulated datasets imitating trajectories in live-cell data [29].

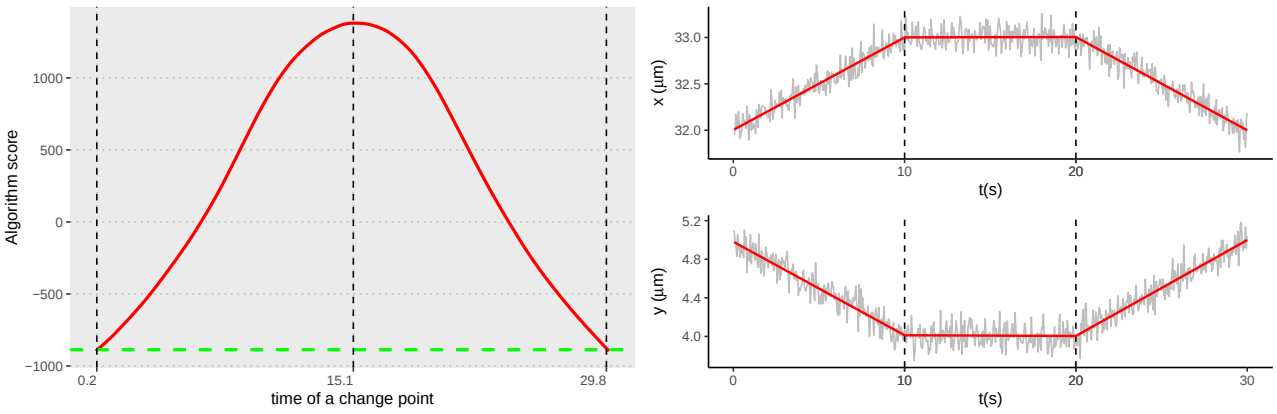
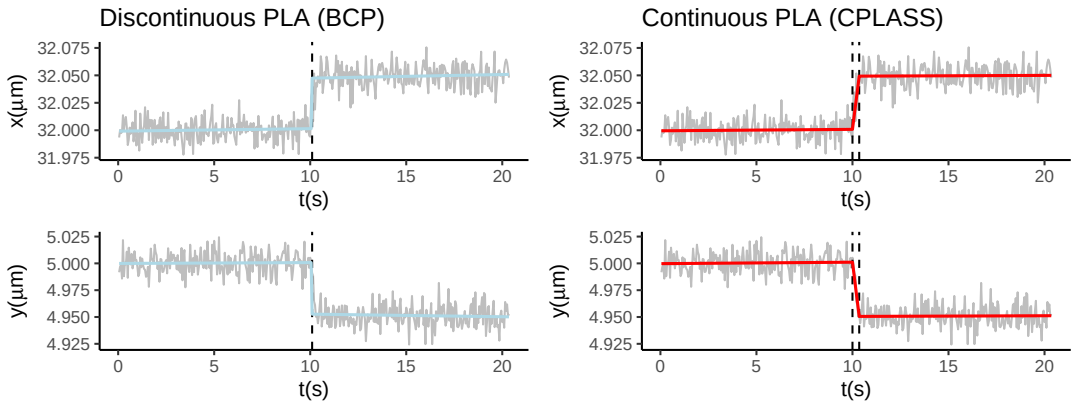


Figure 2. *Failure of binary segmentation in a change-in-velocity setting*. A 2D trajectory was simulated for 30 s at 20 Hz ($\sigma = 0.1$), with true changepoints at 10 and 20 s and segment velocities $(v_x, v_y) = (0.1, -0.1)$, $(0, 0)$, and $(-0.1, 0.1)$ μm/s. **Left**: CPLASS criterion values for the null (green dashed) and one-changepoint models (red solid), showing that binary segmentation introduces a spurious changepoint (higher value is better). **Right**: Observed x and y positions (gray), true changepoints (dashed), and fitted segmentation (red).



Segment	1	2
Duration (s)	10.0500	10.2500
Speed (µm/s)	0.0002	0.0004

Segment	1	2	3
Duration (s)	9.9500	0.3500	10.0000
Speed (µm/s)	0.0002	0.2001	0.0001

Figure 3. *Discontinuous versus continuous piecewise-linear approximations*. A simulated 2D lysosomal trajectory is shown via t–x and t–y time series, with true changepoints at 9 and 10.35 s. Detected changepoints are shown as dashed lines, with the fitted segmentation overlaid in red. The red column in the CPLASS output table indicates a significant active segment missed by BCP.

2. The CPLASS algorithm

In this section, we introduce the statistical model and algorithm used in this work. In Section 2.1, we introduce the statistical model. In Section 2.2, we construct a continuous piecewise-linear approximation of the data given the assumption that the number of changepoints is known. In Sections 2.3 and 2.4, we propose a changepoint detection method where a criterion value with a penalty is provided, and a stochastic search method is used to find the maximum of the defined criterion function.

2.1. Statistical model

We assume that the d-dimensional observations $\{Y_i\}_{i=1}^n \subset \mathbb{R}^d$, collected at times $\mathcal{T} := \{t_1, \ldots, t_n\} \subseteq [0, T]$, are independent Gaussian fluctuations around a sequence of unobserved anchor locations $\{a_i\}_{i=1}^n \subset \mathbb{R}^d$. We write

$$Y_i = a_i + \sigma\varepsilon_i, \tag{2.1}$$

where $\{\varepsilon_i\}_{i=1}^n$ is a sequence of independent and identically distributed d-dimensional standard normal random variables and $\sigma > 0$ is the noise magnitude. We use the conventions $t_0 = 0$ and $a_0 = \underline{a} \in \mathbb{R}^d$.

Assume that the anchor trajectory has k linear segments, with $k + 1 \leq n$, and let τ_j denote the time of the jth change in velocity for $j = 1, \ldots, k-1$. For a natural number k, let $[k] = \{1, \ldots, k\}$. We assume observations are made on a uniform grid of size $\Delta = t_i - t_{i-1} = T/n$ for $i \in [n]$ and that each changepoint is aligned with the observation grid, so that $\tau_j = t_{M_j}$ for some integer $M_j \in \{1, \ldots, n-1\}$. We set $0 =: \tau_0 < \tau_1 < \cdots < \tau_{k-1} < \tau_k := T, M_0 = 0, M_k = n$.

Let $V_j \in \mathbb{R}^d$ denote the velocity vector of the jth segment, with speed $s_j = \|V_j\|_2$, $j \in [k]$, where $\|\cdot\|_2$ is the Euclidean norm. Within segment j, the anchor locations satisfy

$$a_i = a_{M_{j-1}} + V_j(t_i - \tau_{j-1}) \tag{2.2}$$

for $i = M_{j-1} + 1, \ldots, M_j$ and $j \in [k]$.

From the recursive formula (2.2), we construct a multivariate continuous piecewise-linear signal function $f_{\boldsymbol{\tau},\mathbf{V},\underline{a}} : [0, T] \to \mathbb{R}^d$, parametrized by changepoints $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{k-1})$, velocity vectors $\mathbf{V} = (V_1, \ldots, V_k)$, and initial anchor position $\underline{a} \in \mathbb{R}^d$. With the convention $V_0 = 0$, the signal can be written as

$$f_{\boldsymbol{\tau},\mathbf{V},\underline{a}}(t) = \left(\underline{a} - \sum_{m=1}^{j}(V_m - V_{m-1})\tau_{m-1}\right) + V_j t, \qquad t \in [\tau_{j-1}, \tau_j], \quad j \in [k]. \tag{2.3}$$

Equivalently, using the hinge representation,

$$f_{\boldsymbol{\tau},\mathbf{V},\underline{a}}(t) = \underline{a} + V_1 t + \sum_{j=1}^{k-1}(V_{j+1} - V_j)(t - \tau_j)_+, \qquad t \in [0, T], \tag{2.4}$$

where $(x)_+ = x\mathbf{1}\{x > 0\}$.

When $V_{j+1} \neq V_j$ for all $j = 1, \ldots, k-1$, the signal function $f_{\boldsymbol{\tau},\mathbf{V},\underline{a}}$ is said to have k segments and $k-1$ changepoints. Let $\mathcal{F}_k$ denote the collection of signal functions with k segments.

We assume that the observations are generated from a true signal function:

$$Y_i \overset{\text{ind.}}{\sim} \mathcal{N}\left(f^0(t_i), \sigma_0^2 I_d\right), \qquad i = 1, \ldots, n, \tag{2.5}$$

where $f^0 = f_{\boldsymbol{\tau}^0,\mathbf{V}^0,\underline{a}^0}$ is the true signal function with k_0 segments, true changepoints $\boldsymbol{\tau}^0 = (\tau_1^0, \ldots, \tau_{k_0-1}^0)$, true velocities $\mathbf{V}^0 = (V_1^0, \ldots, V_{k_0}^0)$, and initial anchor position $\underline{a}^0 \in \mathbb{R}^d$. Here σ_0^2 is the true variance, and I_d is the d-dimensional identity matrix. Given the observations $(Y_i)_{i=1}^n$, our goal is to infer the true number of segments k_0, the signal parameters $(\boldsymbol{\tau}^0, \mathbf{V}^0, \underline{a}^0)$, and the noise variance σ_0^2.

Given a pre-specified upper bound $\bar{k}$ on the number of segments, we estimate the parameters by maximizing the penalized likelihood:

$$(\widehat{f}_n, \widehat{\sigma}_n^2, \widehat{k}_n) = \underset{\substack{f\in\mathcal{F}_k,\,\sigma^2\in\Omega\\ k\le\bar{k}}}{\arg\max} \left\{ \sum_{i=1}^n \log \mathcal{N}\Big(y_i \mid f_{\boldsymbol{\tau},\mathbf{V},\underline{a}}(t_i), \sigma^2 I_d\Big) - \mathrm{pen}_k \right\}, \tag{2.6}$$

where pen_k is a penalty term to be defined later. Equivalently, for each $k \in [\bar{k}]$, we first compute

$$(\widehat{f}_n^{(k)}, \widehat{\sigma}_{n,k}^2) = \underset{f\in\mathcal{F}_k,\,\sigma^2\in\Omega}{\arg\max} \sum_{i=1}^n \log \mathcal{N}\Big(y_i \mid f_{\boldsymbol{\tau},\mathbf{V},\underline{a}}(t_i), \sigma^2 I_d\Big), \tag{2.7}$$

and then select

$$\widehat{k}_n = \underset{k\in[\bar{k}]}{\arg\max} \left\{ \sum_{i=1}^n \log \mathcal{N}\Big(y_i \mid \widehat{f}_n^{(k)}(t_i), \widehat{\sigma}_{n,k}^2 I_d\Big) - \mathrm{pen}_k \right\}. \tag{2.8}$$

2.2. Continuous piecewise-linear approximation given the changepoints

As noted above, maximizing (2.6) can be separated into two steps. In this section, we address the first, which is to solve (2.7). That is, given a set of fixed changepoints, we derive the maximum likelihood estimates of the continuous piecewise-linear model parameters.

Let $\mathbb{Y} = (y_{il})$ denote the $n \times d$ matrix of observed data. Let $\mathbb{V} = (v_{jl})$ be the $k \times d$ matrix containing all segment velocities, and let $\mathbb{W} = (w_{jl})$ be the $k \times d$ matrix of velocity increments, defined by

$$\mathbb{W}[1, \cdot] = V_1, \qquad \mathbb{W}[j, \cdot] = V_j - V_{j-1}, \quad j = 2, \ldots, k.$$

For fixed changepoints, we estimate the velocity increments w_{jl}, the initial anchor position $\underline{a} = (\underline{a}_1, \ldots, \underline{a}_d)$, and the noise magnitude σ.

Since the d coordinates are conditionally independent given the signal, we can estimate the regression coefficients separately in each dimension. Let

$$Y^{(l)} = (y_{1l}, y_{2l}, \ldots, y_{nl})^\top \in \mathbb{R}^n, \qquad l \in [d]$$

be the lth column of $\mathbb{Y}$, and let

$$\underline{W}^{(l)} = (\underline{a}_l, w_{1l}, \ldots, w_{kl})^\top \in \mathbb{R}^{k+1}$$

contain the lth intercept and velocity-increment coefficients. Then the model in the lth dimension can be written as

$$Y^{(l)} = \mathbb{T}\underline{W}^{(l)} + \sigma\varepsilon^{(l)}, \tag{2.9}$$

where

$$\mathbb{T} = [\,1,\ t,\ (t-\tau_1)_+,\ \ldots,\ (t-\tau_{k-1})_+\,]\Big|_{t=t_i,\ i=1,\ldots,n}, \tag{2.10}$$

with $(x)_+ = x\,\mathbf{1}\{x > 0\}$, $\mathbb{T} \in \mathbb{R}^{n\times(k+1)}$, and $\varepsilon^{(l)} \sim \mathcal{N}(0, I_n)$.

The residual sum of squares (RSS) is

$$\mathrm{RSS}(Y, t; \underline{W}) := \sum_{l=1}^{d} \left\| Y^{(l)} - \mathbb{T}\underline{W}^{(l)} \right\|_2^2. \tag{2.11}$$

The log-likelihood is

$$\begin{aligned}\mathcal{L}(f, \sigma) &= \sum_{i=1}^{n} \log \mathcal{N}\Big(y_i \mid f_{\tau,\mathbf{v},\underline{a}}(t_i), \sigma^2 I_d\Big) \\ &= -\frac{nd}{2}\left[\log(2\pi) + \log(\sigma^2)\right] - \frac{1}{2\sigma^2}\mathrm{RSS}(Y, t; \underline{W}).\end{aligned} \tag{2.12}$$

The resulting maximum likelihood estimators are

$$\widehat{\underline{W}}^{(l)}_{n,k} = (\mathbb{T}^\top\mathbb{T})^{-1}\,\mathbb{T}^\top Y^{(l)}, \qquad l \in [d], \tag{2.13}$$

$$\widehat{\sigma}^2_{n,k} = \frac{\mathrm{RSS}(Y, t; \widehat{\underline{W}}_{n,k})}{dn}. \tag{2.14}$$

As long as the changepoints τ_j are distinct and the design matrix $\mathbb{T}$ has full column rank, $\mathbb{T}^\top\mathbb{T}$ has a unique inverse. From now on, we drop the subscript (n, k) in the maximum likelihood estimators for ease of notation. For the likelihood-based criterion used below, we further restrict the number of segments to $k \le n - 2$ and require $\mathrm{RSS}(Y, t; \widehat{\underline{W}}) > 0$. Since $\mathbb{T}$ has $k + 1$ columns, $k \le n - 2$ excludes the saturated $n \times n$ case. Together with the full-rank condition and positive RSS, this ensures that $\widehat{\sigma}^2_{n,k} > 0$ and that the maximized log-likelihood is finite.

The velocity vector corresponding to the jth segment is recovered from the velocity increments as

$$\widehat{V}_j = \left(\sum_{r=1}^{j} \widehat{w}_{r1}, \ldots, \sum_{r=1}^{j} \widehat{w}_{rd}\right)^\top, \qquad j \in [k].$$

The speed of the associated segment is $\widehat{s}_j = \|\widehat{V}_j\|_2$.

Biologically, each changepoint represents a change in the particle's multidimensional velocity vector, while the duration of the corresponding segment quantifies how long that kinematic regime persists. Transitions between near-zero and nonzero inferred speeds are consistent with transitions between near-stationary behavior and directed motion, whereas changes in velocity magnitude or direction may reflect pauses, reversals, altered motor activity, or interactions with the cellular environment. These interpretations are kinematic rather than uniquely mechanistic: Trajectory data alone cannot determine whether a near-stationary segment results from motor stalling, detachment, opposing motor forces, or confinement.

2.3. Criterion function

The typical representation of the model contains changepoint times $\boldsymbol{\tau}$, initial intercepts (or initial anchor locations) $\underline{a}$, segment velocities $\mathbf{V}$, and noise σ. In the following, we introduce a reduced representation of the model that uses a changepoint vector $r = (r_1, \ldots, r_{n-2})$ of ones and zeros, in the spirit of [30]. When $r_i = 1$, this indicates that a changepoint occurs at the corresponding internal observation-grid location.

Henceforth we use the notation $|r|$ to denote the number of changepoints, so that the corresponding number of segments is $K(r) = |r| + 1$.

We restrict the stochastic search to an admissible changepoint state space. Let $\bar{k}$ be the practitioner-specified upper bound on the number of segments introduced in Eq (2.6), and let $L_{\min} \geq 1$ denote the minimum permitted spacing between adjacent segment boundaries in observation-index units. For a changepoint configuration r, let $1 = M_0 < M_1 < \cdots < M_{|r|} < M_{|r|+1} = n$ denote its ordered segment boundaries on the observation grid. We define

$$\mathcal{R}_n = \left\{ r \in \{0,1\}^{n-2} : |r| \leq n-3, |r|+1 \leq \bar{k}, M_j - M_{j-1} \geq L_{\min} \text{ for all } j, \operatorname{rank}(\mathbb{T}_r) = |r|+2, \widehat{\mathrm{RSS}}_r > 0 \right\}.$$

Thus the effective upper bound on the number of segments is $K(r) \leq \min\{\bar{k}, n-2\}$. In our implementation, $L_{\min} = 1$ is the default, corresponding to the weakest spacing restriction, although the practitioner may specify a larger value.

For any given a particular changepoint vector r, we can find an associated piecewise-linear approximation whose $\mathrm{RSS}(Y, t; \widehat{\underline{a}}, \widehat{W}), \tau_j, \hat{s}_j$ ($j \in [|r|+1]$) are determined as discussed in Section 2.2. To this end, we will use the subscript r to indicate these relationships, i.e., $\widehat{\mathrm{RSS}}_r = \mathrm{RSS}(Y, t; \widehat{\underline{a}}_r, \widehat{W}_r), \tau_j, \hat{s}_j, \widehat{\underline{a}}_r, \tau_{j,r}, \hat{s}_{j,r}, \widehat{V}_{j,r}$. Altogether, for a given r, according to the derivation in Section 2.2, the piecewise-linear model yields a maximized value of the log-likelihood function when it has the form

$$\widehat{\mathcal{L}}_n = \log(L(Y, t; \widehat{\underline{a}}_r, \widehat{W}_r, \hat{\sigma}_r)) = \log\left(\frac{1}{(2\pi)^{nd/2}} \left(\frac{dn}{\widehat{\mathrm{RSS}}_r} \right)^{nd/2} \exp\left(-\frac{nd}{2} \right) \right) \tag{2.15}$$

$$= -\frac{nd}{2} \log\left(\widehat{\mathrm{RSS}}_r \right) + C, \tag{2.16}$$

where C is a constant. We define the criterion function of the algorithm as follows.

Definition 1 (Form of the criterion function)**.**

$$\Phi(r) = 2\widehat{\mathcal{L}}_n - \mathrm{pen}(r) = -nd \log\left(\widehat{\mathrm{RSS}}_r \right) - \mathrm{pen}(r). \tag{2.17}$$

Here, $\mathrm{pen}(r)$ refers to the penalty term designed to prevent overfitting. We used an sSIC penalty expressed as $(\log n)^\gamma \rho$, where ρ is the number of parameters in the model and requires $\gamma > 1$ (see [15,31]) and a speed-control penalty function to mitigate the occurrence of unrealistic speed values.

Definition 2 (Form of the penalty function)**.**

$$\mathrm{pen}(r) = (\log n)^\gamma \rho + \eta \sum_{j=1}^{K(r)} h\left(\frac{\hat{s}_{j,r}}{s_{cap}} - 1 \right), \tag{2.18}$$

where $\gamma > 1$, $\rho = |r|(d+1) + 2d + 1 = K(r)(d+1) + d$ *is the total number of model parameters,* $|r|$ *is the number of changepoints,* $K(r) = |r| + 1$ *is the number of segments, and* $\widehat{s}_{j,r} = \left\|\widehat{V}_j\right\|_2$, $j \in [K(r)]$ *is the estimated speed of segment* j*. The quantity* $s_{\text{cap}} > 0$ *is a physically specified reference speed above which soft regularization begins,* $\eta \geq 0$ *is a dimensionless coefficient controlling the strength of that regularization, and* $h(s) = \max\{0, s\}$.

The speed regularizer incorporates prior scientific knowledge about biologically realistic particle speeds. The value s_{cap} is not a hard upper bound: Segments with fitted speeds above it remain admissible when sufficiently supported by the data. If no reliable reference speed is available or speed regularization is not desired, the practitioner may set $\eta = 0$, in which case the penalty reduces to the sSIC term [15, 31]. We remark that $\gamma = 1$ corresponds to the standard Schwarz/Bayesian information criterion (SIC/BIC) penalty considered by [32] in the context of multiple changepoint detection. In CPLASS, we adopt $\gamma > 1$ to preserve the formal consistency guarantees developed in the author's doctoral dissertation [33] and in a companion methodological manuscript [34]. This strengthened penalty has also been successfully used in wild binary segmentation [31] and the NOT algorithm [15]. Based on the sensitivity analysis in Section 3.1.1 and Figure 6, we recommend $\gamma = 1.01$ when sensitivity to short motile segments is important. For the optional speed regularizer, we use $\eta = 1$ as a moderate default; its calibration and sensitivity are examined in Supplementary Section S.7. Both parameters remain user-adjustable, and setting $\eta = 0$ recovers the sSIC-only criterion.

2.4. Stochastic search of the changepoint space

When it comes to the search methods, the issue with using popular methods such as binary segmentation, PELT, or optimal partitioning has been discussed in the introduction. Another popular approach, called gradient ascent, has been used in optimization problems. However, this is not an ideal strategy because our criterion function commonly has multiple local maxima (see Figure 4 for an example). We chose a stochastic search approach using a Metropolis–Hastings (MH) algorithm as the search algorithm.

2.4.1. Metropolis–Hastings algorithm as the searching algorithm

Notice that finding the maximum of the $\Phi(r)$ function is equivalent to finding the maximum of the function $\exp(\Phi(r))$. Using MH, we can generate an ergodic Markov chain $\{r^{(t)}\}_{t\geq 0}$ that has $\pi(r) = C\exp\{\Phi(r)\}$ as its target stationary distribution on the admissible changepoint state space $\mathcal{R}_n$, where C is a normalizing constant. The maximum of the $\exp(\Phi(r))$ function can then be approximated by the maximum of the sequence $\{\exp(\Phi(r^{(t)}))\}$.

As defined above, $\mathcal{R}_n$ contains only changepoint configurations for which the continuous piecewise-linear model is well defined. In particular, configurations in $\mathcal{R}_n$ satisfy the prescribed maximum-number-of-segments and minimum-segment-length restrictions, $|r| \leq n - 3$, the corresponding design matrix has full column rank, and the residual sum of squares is positive. Therefore, $\Phi(r)$ is finite for every $r \in \mathcal{R}_n$.

Because the independent proposal described below assigns positive probability to every admissible changepoint configuration, the resulting chain is irreducible. Moreover, the chain has positive self-transition probability and hence is aperiodic. Since $\{r^{(t)}\}_{t\geq 0}$ takes its values in the finite space $\mathcal{R}_n$, it is uniformly ergodic [35]. Consequently, as the number of MH iterations tends to infinity, the probability that the best state visited by the chain is a global maximizer of $\Phi(r)$ tends to one. This asymptotic

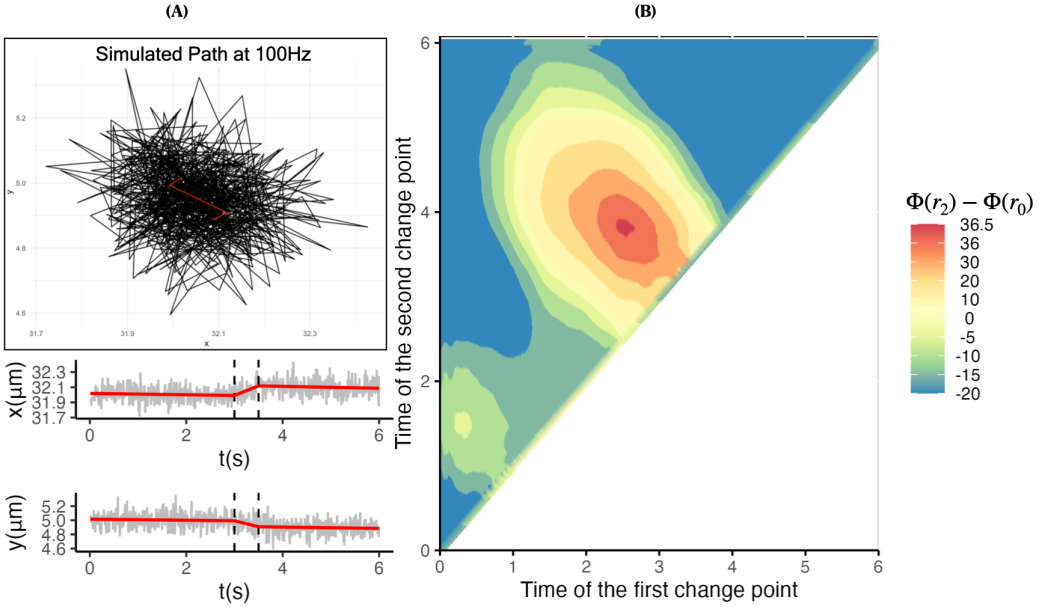


Figure 4. *Example showing why gradient ascent is unsuitable for the change-in-speed problem.* **(A)** A simulated 2D lysosomal trajectory (100 Hz, 6 s) with true changepoints at 3 and 3.5 s and segment speeds $(0, 0.2, 0)\,\mu$m/s. Noisy t–x and t–y time series are shown in gray, with true changepoints (dashed) and the true segmentation (red). **(B)** Contours of CPLASS score differences between the two-changepoint model $\Phi(r_2)$ and the no-changepoint model $\Phi(r_0)$. Although the global maximum occurs at the true changepoints, gradient ascent can become trapped in local maxima and fail to recover them.

statement does not guarantee that a global maximizer is reached under a fixed computational budget.

In Supplementary Section S.3, we show that the detailed balance condition holds for our proposed rules.

To be specific, in our algorithm, the proposal function for the changepoint process takes its values on the admissible space $\mathcal{R}_n \subseteq \{0,1\}^{n-2}$, where $n-2$ is the number of internal candidate changepoint locations. There are four types of changepoint vector proposals: (1) an independent changepoint vector; (2) the creation or extinction of a changepoint; (3) the creation or extinction of a segment; or (4) a location shift of a single changepoint. Let r^{prop} and r^{cur} denote the proposed and current changepoint process, respectively. The following summary describes each type of proposal.

- **Type 1.** Notation q_{new}. Including this type of proposal allows the changepoint space walk to escape local maxima and for the number of changepoints to vary by proposing an independent changepoint vector r^{prop}. Let $p_\lambda = 1 - \exp(-\lambda\Delta)$. We independently generate $r_i^{\text{prop}} \overset{iid}{\sim} \text{Bernoulli}(p_\lambda), i \in [n-2]$. For an admissible proposed configuration,

$$\begin{aligned} q_{\text{new}}(r^{\text{prop}}|r^{\text{cur}}) &= q_{\text{new}}(r^{\text{prop}}) \\ &= (1-\exp(-\lambda\Delta))^{|r^{\text{prop}}|} \times \exp(-\lambda\Delta)^{n-2-|r^{\text{prop}}|}. \end{aligned}$$

  If the generated configuration is outside $\mathcal{R}_n$, the proposal is treated as a self-transition, and the chain remains at r^{cur}.

Here λ is chosen by the practitioner to qualitatively match the number of changepoints that is expected in the data set. Without calibration to realistic numbers of changes, proposals of this type will tend to be rejected.

- **Type 2.** Notation q_{bd}. While Type 1 allows us to independently draw a new changepoint vector r^{prop}, the second type proposal, q_{bd}, provides another way for the number of changepoints to vary from iteration to iteration given the current changepoint vector r^{cur}. Instead of using the proposal function q_{bd} mentioned in [36], we modify it such that the new version of q_{bd} will add one or delete one changepoint on the current list of changepoints with equal probabilities.
  Let $\mathbf{M}_{r^{\text{cur}}} = \{i : r_i^{\text{cur}} = 1\}$ be the set of current changepoint indices among the N internal candidate locations, and let $r^{\text{prop}} = r^{\text{cur}}$. There is a 50% chance that a component is randomly sampled from the current changepoint indices,
  $$s \sim \text{Uniform}(\mathbf{M}_{r^{\text{cur}}}),$$
  and we let $r_s^{\text{prop}} = 0$. Otherwise, a component s is drawn from the complement of the current changepoint indices,
  $$s \sim \text{Uniform}(\mathbf{M}^c_{r^{\text{cur}}}),$$
  and we let $r_s^{\text{prop}} = 1$.
  For an admissible state-to-state proposal,
  $$q_{\text{bd}}(r^{\text{prop}}|r^{\text{cur}}) = \begin{cases} \dfrac{1}{2|r^{\text{cur}}|}, & \text{if we remove a changepoint,} \\ \dfrac{1}{2(n-2-|r^{\text{cur}}|)}, & \text{if we add a changepoint.} \end{cases}$$
  The birth and death probabilities remain equal to $1/2$ at the boundaries. If the selected direction is unavailable, or if the resulting configuration is outside $\mathcal{R}_n$, the move is a self-transition.
- **Type 3.** Notation q_{bd_2}. The third type of proposal, q_{bd_2}, allows for adding or removing two nearby changes in the current list of changepoints. Like Type 2, this proposal also provides a way to vary the number of changepoints in each iteration. In Section 2.4.2, we provide motivation for including and removing consecutive changes. Let $r^{\text{prop}} = r^{\text{cur}}$. We set the chances of adding or deleting a segment to be equal.
  We first provide the changepoint insertion procedure in detail. Let $M_1 < \cdots < M_{|r^{\text{cur}}|}$ denote the ordered changepoint locations in r^{cur}, and let $M_0 = 1$ and $M_{|r^{\text{cur}}|+1} = n$ denote the two trajectory boundaries. For each current segment j, define
  $$B_j(r^{\text{cur}}) = \left\{\{s, s'\} : M_{j-1} < s < s' < M_j,\ r^{+(s,s')} \in \mathcal{R}_n\right\},$$
  where $r^{+(s,s')}$ denotes the configuration obtained by adding changepoints at s and s'. Thus, the two proposed changepoints are chosen from the strict interior of the segment and cannot coincide with an existing segment boundary. Let $E_B(r^{\text{cur}}) = \{j : B_j(r^{\text{cur}}) \neq \varnothing\}$ be the set of current segments in which two admissible changepoints can be inserted.
  In the case where we add a segment, one segment is first sampled uniformly from $E_B(r^{\text{cur}})$, and then one pair $\{s, s'\}$ is sampled uniformly from $B_j(r^{\text{cur}})$. For the particular state r^{prop} generated by this pair,
  $$q_{\text{bd}_2}(r^{\text{prop}}|r^{\text{cur}}) = \frac{1}{2}\frac{1}{|E_B(r^{\text{cur}})|}\frac{1}{|B_j(r^{\text{cur}})|}.$$

For deletion, define the set of consecutive changepoints, i.e., adjacent changepoints in the ordered changepoint list, to be

$$D(r^{\text{cur}}) = \left\{ \{M_j, M_{j+1}\} : j = 1, \ldots, |r^{\text{cur}}| - 1,\ r^{-(M_j, M_{j+1})} \in \mathcal{R}_n \right\},$$

where $r^{-(M_j, M_{j+1})}$ denotes the configuration obtained by deleting the two changepoints. If we remove a segment, one pair is sampled uniformly from $D(r^{\text{cur}})$. For the particular resulting state,

$$q_{\text{bd}_2}(r^{\text{prop}} | r^{\text{cur}}) = \frac{1}{2} \frac{1}{|D(r^{\text{cur}})|} = \frac{1}{2(|r^{\text{cur}}| - 1)}.$$

If birth is selected and $E_B(r^{\text{cur}}) = \varnothing$, or if death is selected and $D(r^{\text{cur}}) = \varnothing$, the proposal is a self-transition. Thus the birth/death probabilities remain $1/2$ at all states and are not redistributed at the boundaries.

- **Type 4.** Notation q_{shift}. This type of proposal allows exploration of the best combination of changepoints for a fixed number of changepoints. We obtain the proposed changepoint vector by randomly sampling two components of the current changepoint vector as follows:

$$s \sim \text{Uniform}\left(\mathbf{M}_{r^{\text{cur}}}\right),$$
$$s' \sim \text{Uniform}\left(\mathbf{M}^c_{r^{\text{cur}}}\right).$$

The proposal of this type is defined as

$$r_i^{\text{prop}} = \begin{cases} r_i^{\text{cur}}, & \text{if } i \neq s, s', \\ 1 - r_i^{\text{cur}}, & \text{otherwise.} \end{cases}$$

For an admissible proposed state,

$$q_{\text{shift}}(r^{\text{prop}} | r^{\text{cur}}) = \frac{1}{|r^{\text{cur}}|} \times \frac{1}{N - |r^{\text{cur}}|}.$$

Since $|r^{\text{prop}}| = |r^{\text{cur}}|$, the reverse move has the same probability, so this proposal is symmetric. If the proposed shift produces a configuration outside $\mathcal{R}_n$, it is treated as a self-transition.

Finally, we combine all these proposal types into one proposal mechanism. At each iteration, we draw $u_r \sim \text{Uniform}(0, 1)$. According to the interval containing u_r, exactly one of the four proposal types is selected and applied:

$$q_r(r^{\text{prop}} | r^{\text{cur}}; u_r) = \begin{cases} q_{\text{new}}(r^{\text{prop}} | r^{\text{cur}}), & 0 \leq u_r \leq u_1, \\ q_{\text{bd}}(r^{\text{prop}} | r^{\text{cur}}), & u_1 < u_r \leq u_2, \\ q_{\text{bd}_2}(r^{\text{prop}} | r^{\text{cur}}), & u_2 < u_r \leq u_3, \\ q_{\text{shift}}(r^{\text{prop}} | r^{\text{cur}}), & u_3 < u_r \leq 1. \end{cases} \tag{2.19}$$

We then generate $r^{\text{prop}} \sim q_r(\cdot\,| r^{\text{cur}}; u_r)$.

The quantities

$$u_1, \qquad u_2 - u_1, \qquad u_3 - u_2, \qquad 1 - u_3$$

are the probabilities that proposal Types 1, 2, 3, and 4 are chosen, respectively. We set

$$u_1 = \frac{1}{4}, \qquad u_2 = \frac{3}{8}, \qquad u_3 = \frac{1}{2}$$

as default in the algorithm.

These proposal-type probabilities are fixed and do not depend on the current changepoint configuration. Exactly one proposal type is selected and applied in each MH iteration. Let P_ℓ denote the complete MH transition kernel corresponding to proposal Type ℓ, $\ell = 1, \ldots, 4$. The marginal transition kernel of the search can then be written as $P(r^{\text{prop}}|r^{\text{cur}}) = \sum_{\ell=1}^{4} w_\ell P_\ell(r^{\text{prop}}|r^{\text{cur}})$, where $(w_1, w_2, w_3, w_4) = \left(\frac{1}{4}, \frac{1}{8}, \frac{1}{8}, \frac{1}{2}\right)$. This expression averages over the randomly selected proposal type; only one proposal type is applied at each iteration. In Supplementary Section S.3, we show that each P_ℓ satisfies detailed balance with respect to π, and therefore their fixed mixture also satisfies detailed balance.

The sampling algorithm is then described in Algorithm 1. We then introduce the CPLASS algorithm as in Algorithm 2.

Algorithm 1 MH algorithm: Unknown number of changepoints

Input: The observed data $(\mathbf{x}, \mathbf{y}, \mathbf{t})$, the rate of changepoint processes (λ), and a time step (Δ). The number of iterations $(T_{\max})$.

Output: A list containing $T_{\max}$ changepoint vectors $\{r^{(t)}\}_{t=0}^{T_{\max}}$ with their corresponding $\widehat{\text{RSS}}_r(\mathbf{x}, \mathbf{y}, \mathbf{t}; r^{(t)})$, $\tau_{k,r^{(t)}}$, and $\hat{s}_{k,r^{(t)}}$ for $k \in [K_{r^{(t)}}]$.

$t = 0$. Draw an initial admissible state $r^{(0)} \in \mathcal{R}_n$ from the Bernoulli proposal distribution.
Compute $\widehat{\text{RSS}}_r(\mathbf{x}, \mathbf{y}, \mathbf{t}; r^{(0)})$, $\tau_{k,r^{(0)}}$, and $\hat{s}_{k,r^{(0)}}$ (for $k \in [K_{r^{(0)}}]$) using the piecewise-linear approximations.
for $t = 1, 2, \ldots, T_{\max}$ **do**
    Draw $u_r \sim \text{Uniform}(0, 1)$ and determine $q_r(\cdot|\cdot; u_r)$ according to Equation (2.19).
    Generate

$$r^{\text{prop}} \sim q_r(\,\cdot\,|r^{(t-1);u_r}).$$

    if $r^{\text{prop}} \neq r^{(t-1)}$ **then**
        Compute the acceptance probability

$$\log\left(\alpha_{K_t}\left(r^{(t-1)}, r^{\text{prop}}\right)\right) = \min\left\{0,\ \Phi(r^{\text{prop}}) - \Phi(r^{(t-1)}) + \log\left(\frac{q_r(r^{(t-1)}|r^{\text{prop}})}{q_r(r^{\text{prop}}|r^{(t-1)})}\right)\right\}.$$

        if $\alpha_r(r^{(t-1)}, r^{\text{prop}}) \geq \text{Uniform}(0, 1)$ **then**
            Set $r^{(t)} = r^{\text{prop}}$.
        else
            Set $r^{(t)} = r^{(t-1)}$.
        end if
    else
        Set $r^{(t)} = r^{(t-1)}$.
    end if
    Compute $\widehat{\text{RSS}}_r(\mathbf{x}, \mathbf{y}, \mathbf{t}; r^{(t)})$, $\tau_{k,r^{(t)}}$, and $\hat{s}_{k,r^{(t)}}$ (for $k \in [K_{r^{(t)}}]$) using the piecewise-linear approximations.
end for

2.4.2. Necessity of Type 3 proposals

While it is natural to add, subtract, or shift individual points, the specific structure of the change-in-velocity model, which may involve biologically significant short, fast segments, also requires proposals

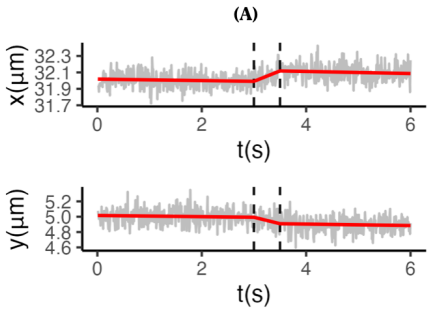
Algorithm 2 CPLASS algorithm

Input: The output from running Algorithm 1.
Output: A list containing the continuous piecewise-linear approximation of the data in terms of **x** and **y**, changepoint times, segment durations, and segment speeds.

1: Find the maximum of the collected $\{\Phi(r^{(t)})\}_{t=1,\ldots,T_{\max}}$ and return the corresponding $r^{(t^*)}$.
2: Using the continuous piecewise-linear approximation with the selected $r^{(t^*)}$, return the final output of the algorithm.

for adding or subtracting pairs of changepoints that would mark the endpoints of short segments. Certainly, MH will still succeed in the absence of such proposals, but finding short, fast segments can take much longer. This can happen because it is unlikely in long paths for the algorithm to randomly propose two positions that are close to each other, but there is a deeper issue that also impedes convergence. Similar to what is seen in Figure 2, adding one endpoint of a short segment, but not the other, can lead to a decrease in the criterion function. So it becomes unlikely that a correct proposal of one short segment endpoint will be accepted without a companion endpoint.

We demonstrate the issue explicitly with the following numerical experiment. We simulated a 6-second path at 100 Hz with changes at $t = 3$ s and $t = 3.5$ s. See Figure 5(A). The segment speeds are $(0, 0.2, 0)$ μm/s, and we assume that the current changepoint vector is $r^{\text{cur}} = \mathbf{0}$, i.e., there are no changepoints. Figure 5(B) provides the corresponding log-likelihood and criterion values for three proposals: two that involve adding one of the true endpoints of the active segment, and one that adds both simultaneously. Given the displayed criterion function evaluations, a progression from zero inferred changepoints to one and then two is prohibitively unlikely.



(B)

Changepoints	r	Log_likelihood	Criterion Value	Decision
No changepoints	..00000...00000...	1028.784	2024.983	
At t = 3	..00010...00000...	1031.056	2009.977	✖ Reject
At t=3.5	..00000...01000...	1028.846	2005.558	✖ Reject
At t=3, t=3.5	..00010...01000...	1063.836	2055.988	✔ Accept

Figure 5. *Necessity of the new proposal function* (Numerical Experiment 2.4.2). **(A)** A simulated 2D lysosomal trajectory (100 Hz, 6 s) with true changepoints at 3 and 3.5 s, shown in t–x and t–y coordinates. True changepoints are dashed; the fitted segmentation is overlaid in red. **(B)** Candidate changepoint models with their log-likelihoods, criterion values, and corresponding MH decisions (Algorithm 1).

2.4.3. Computational considerations

CPLASS uses MH proposals to search for the changepoint configuration maximizing the penalized criterion. The resulting chain is not used for posterior inference. We therefore assess stabilization of the running maximum and agreement across independent starts rather than posterior mixing. For fixed $K_{\max}$, the computational cost is approximately linear in trajectory length per iteration, up to matrix-factorization and proposal costs, and independent starts can be evaluated in parallel. Benchmarks of

runtime, early stopping, multistart agreement, and criterion stabilization are reported in Supplementary Section S.5.

3. Results

This section evaluates the practical performance of CPLASS on biologically motivated simulations and on experimental intracellular transport datasets.

3.1. Numerical experiments on the choice of the penalty term and comparison with BCP

Now we validate the CPLASS algorithm and examine the penalty terms through simulation studies. Section 3.1.1 illustrates the performance of CPLASS when varying the value of γ in the linear penalty term. Section 3.1.2 demonstrates that CPLASS can detect short, fast segments better than BCP. However, in the presence of noise, the algorithm can sometimes detect a too-fast and too-short segment, leading to large speeds that are biophysically implausible. We will show that incorporating the speed penalty can help avoid this phenomenon (by slightly shifting the changepoint positions) without affecting the estimated number of changes or the overall signal function.

3.1.1. Sensitivity to the model-selection penalty exponent

We evaluated the sensitivity of CPLASS to the exponent γ in the model-selection penalty using $\gamma \in \{1.01, 1.05, \ldots, 2.00\}$ (Figure 6). We selected $n = 53$ to represent a practically relevant short trajectory, for which strong penalization can substantially reduce detection power, and $n = 203$ to test detection of an extremely brief three-interval motile segment within a longer record. We compared H_0, with no changepoints, against H_1, with two closely spaced changepoints. For each path length and hypothesis, we analyzed the same 200 simulated trajectories at every value of γ, both with and without the speed penalty.

For $n = 53$ (20 Hz, 2.65 s), the changepoints under H_1 occurred at 1.10 and 1.55 s, so the motile segment spanned nine sampling intervals; the segment speeds were $(0, 0.10, 0)\,\mu$m/s. For $n = 203$ (20 Hz, 10.15 s), the changepoints occurred at 5.00 and 5.15 s, so the motile segment spanned only three sampling intervals; the segment speeds were $(0, 0.15, 0)\,\mu$m/s. In all cases, $\sigma = 0.01\,\mu$m. Performance was measured by the probability of selecting exactly two changepoints under H_1 and the probability of selecting any changepoint under H_0.

At $\gamma = 1.01$, CPLASS selected two changepoints in more than 90% of trajectories for both path lengths while maintaining a false-positive probability below 5%. Increasing γ progressively reduced detection of the short motile segment, particularly for $n = 53$, without meaningfully improving false-positive control. We therefore recommend $\gamma = 1.01$ as the default when sensitivity to short motile segments is important. Results with and without the speed penalty were nearly identical in these low-speed simulations, indicating that the speed penalty had no material effect in this setting.

3.1.2. Effect of the speed penalty

We investigated CPLASS and BCP's performance on a collection of 250 simulated trajectories derived from the base parameter sets in Table 1 from [29] (also see Table S2 in Supplementary Section S.2) at 25 Hz, which is designed to mimic real trajectories. Recall that BCP models change-in-mean of

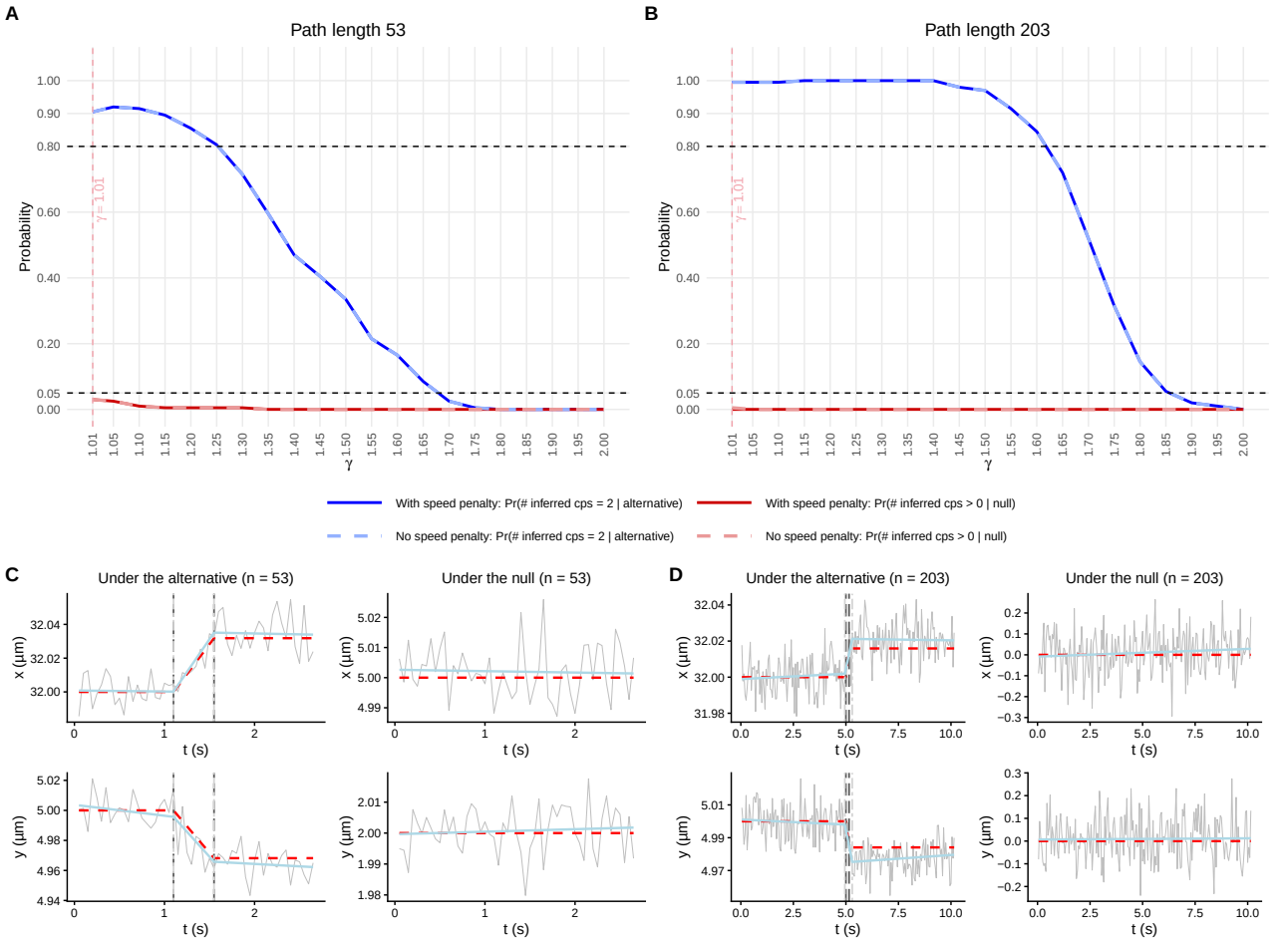


Figure 6. *Sensitivity of CPLASS to* γ. **(A)** Detection probabilities for $n = 53$. **(B)** Corresponding results for $n = 203$. Blue curves show $\Pr(\widehat{|r|} = 2 \mid H_1)$, and red curves show $\Pr(\widehat{|r|} > 0 \mid H_0)$. Dark solid and light dashed curves denote CPLASS with and without the speed penalty, respectively. The vertical line marks $\gamma = 1.01$; horizontal lines mark probabilities 0.80 and 0.05. **(C,D)** Representative trajectories (path 1) for $n = 53$ and $n = 203$. Gray curves show observations, red dashed curves show the generating trajectories, and blue curves show CPLASS fits. Gray long-dashed and black double-dashed vertical lines mark inferred and true changepoints, respectively.

location increments while CPLASS directly models the trajectories as piecewise-linear functions. We compared the inferred segmentation outputs by BCP, CPLASS (with and without speed penalties) and the truth by using two summary statistics, namely the CSA statistic introduced in [29] and the cumulative distribution function (CDF) of the inferred maximum segment speeds (see Figure 7). Briefly, for every speed $s \geq 0$, the CSA is the inferred proportion of time spent at speeds less than or equal to s. We refer to [29] for a more detailed discussion on the CSA.

In Panel (A) of Figure 7, we display the result of applying the CPLASS (with and without the speed penalty) and BCP algorithms to 250 simulated trajectories. Each member of the CSA curve ensembles, orange for the BCP output, blue for CPLASS without the speed penalty, and green for CPLASS with the

speed penalty, is the inferred CSA calculated from bootstrap resampling of the 250 paths. The evident gap between the CSA ensembles highlights the distinction between BCP and CPLASS, particularly in the proportion of time that the simulated particles are moving at speeds of 0.5 μm/s or slower. Meanwhile, both versions of CPLASS (with the speed penalty activated or deactivated) closely follow the theoretical CSA curve (in black), which was used to simulate the data. This is consistent with the argument we discussed earlier in this paper regarding the issue of missing short, fast segments when using the discontinuous piecewise-linear model.

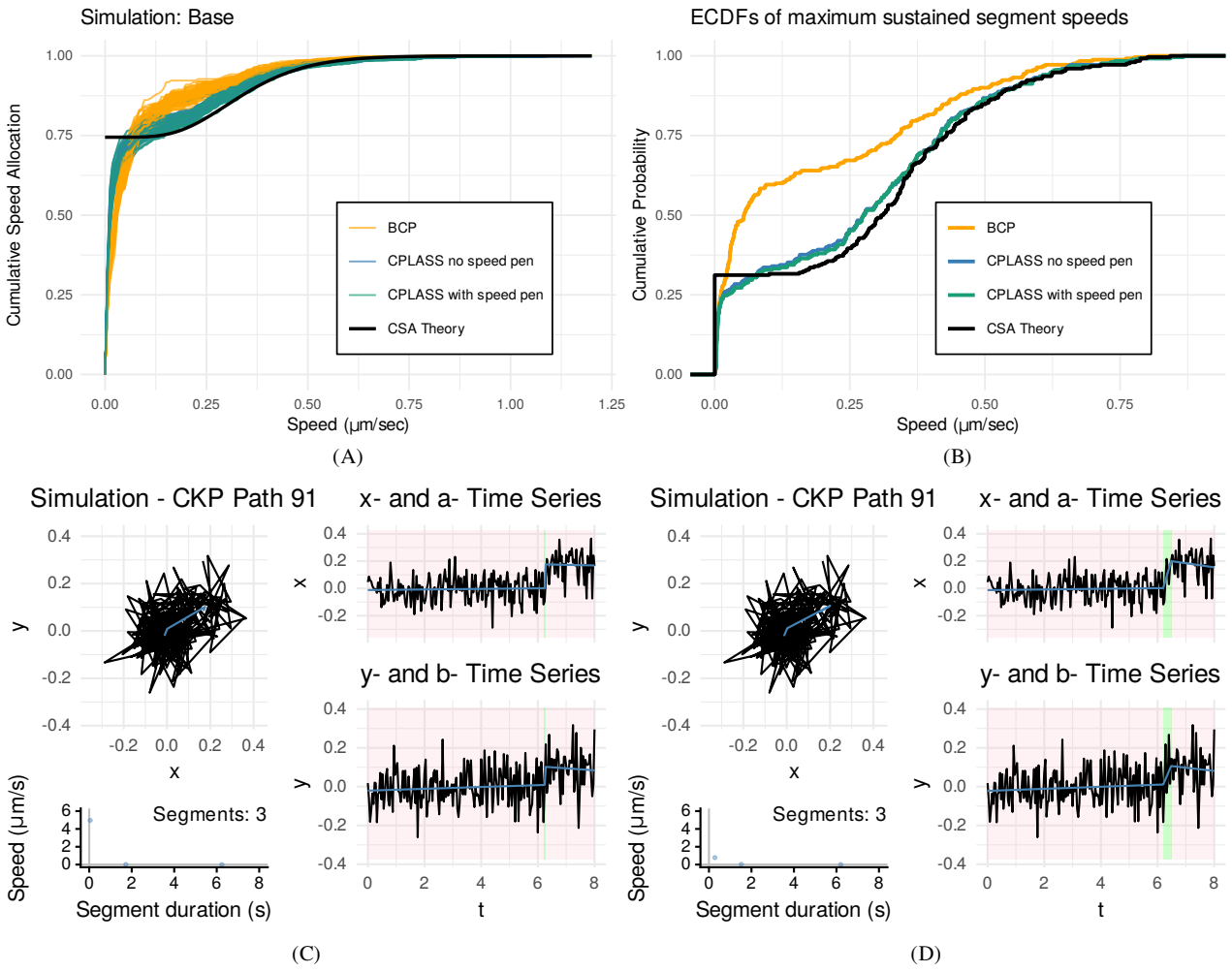


Figure 7. *Necessity of the speed penalty (Section 3.1.2).* We simulated 250 trajectories from the parameter sets in Table 1 of [29], sampled at 25 Hz. (**A**) CSA: theoretical CSA (black), CPLASS with speed penalty (green), CPLASS without speed penalty (blue), and BCP [25] (orange). (**B**) Empirical cumulative distributions of maximum sustained segment speeds (minimum duration 0.6 s) for BCP (orange), CPLASS with (green) and without (blue) the speed penalty, compared with the true values (black). (**C,D**) A representative trajectory after running CPLASS without/with speed penalty.

The right panel of Figure 7 illustrates the empirical cumulative distribution (ECDF) of the *maximum sustained segment speeds* after running the BCP and the two versions of CPLASS. Here we present

the ECDF of the maximum segment speeds where the associated durations are at least 0.6 s, since very short durations make it difficult to detect changepoints. We observed that CPLASS, with or without the speed penalty, more accurately matches the CDF of the true maximum speed than the results produced by BCP. It is not immediately obvious from the ECDF or CSA that adding a speed penalty makes a difference, since both versions of the method approximate the true function equally well at the global level. However, differences become apparent in specific trajectories, such as those shown in Panels (C) and (D). In these cases, CPLASS with the speed penalty adjusts the estimated speed of short, fast segments from approximately 4.961 to 0.786 μm/s by slightly widening the estimated segment. Using the reference speed $s_{\text{cap}} = 1\,\mu$m/s, we can see how the speed penalty softly regularizes the segment speed toward values below this threshold.

We additionally assessed sensitivity to the dimensionless regularization coefficient η. In the low-speed null and alternative simulations used for the γ analysis, the selected fitted speeds remained below s_{cap}, and varying η had no effect on the detection results. We then examined the fixed subset of 26 Cook-simulation trajectories whose unpenalized selected fits contained speeds above s_{cap}. Increasing η progressively reduced these extreme fitted speeds: The median trajectory-level maximum speed decreased from 2.42 μm/s at $\eta = 0$ to 1.02 μm/s at $\eta = 1$, while the overall maximum decreased from 5.49 to 2.36 μm/s. Stronger values produced smaller additional reductions and altered more inferred changepoint locations. We therefore use $\eta = 1$ as a moderate default while treating it as a user-adjustable regularization coefficient. Complete sensitivity results are reported in Supplementary Section S.7.

3.2. Change-in-mean versus change-in-velocity analysis

Here we report the results of a direct power comparison between change-in-mean analysis of location increments (using BCP) versus continuous piecewise-linear approximation (CPLASS with and without speed penalty) by simulating 20,000 paths under the same conditions. All paths had two actual changes observed at 20 Hz with $\sigma = 0.01$, but the positions of the changepoints varied due to differences in the duration and speed of the middle segment. The 20 values for the middle segment duration were selected over the interval from 0.05 to 1 s, with an increment of 0.05 s. The middle segment speed values (20 different values in μm/s) ranged from 0.01 to 0.2 μm/s, increasing by 0.01 μm/s. We fixed the durations of the first and third segments (2 s) as well as those segment speeds (0 μm/s). Therefore, the sample sizes (number of location observations) varied from $n = 81$ to 100. For each variation of the pair of speed and duration, there were 50 corresponding simulated paths. The correctly detected percentage $\mathbf{P}_{\text{correct}}$ was then computed for each speed/duration pair as the proportion of paths in which the segmentation algorithm reported the correct number of changes. We ran CPLASS and BCP for these 20, 000 simulation paths; the results showed that CPLASS (with both versions) was better than BCP in detecting short segments and slow segments, which BCP treated as having no movement (e.g., see Supplementary Section S.1.2, Figure S3). Figure 8 illustrates the comparison. Regions in which the algorithms are considered “effective" are colored in green (see legend for color gradient). The region in which CPLASS (both with and without the speed penalty) is effective is notably larger than that of BCP. CPLASS also achieved better performance than BCP in detecting the true number of changepoints with $\mathbf{P}_{\text{correct}} \geq 90\%$ for all cases where the middle segment had a duration varying from 0.45 to 1 s and the speed varying from 0.08 to 0.2 μm/s. Additionally, this experiment again confirms that adding the speed penalty function to the penalty function maintains the correctly detected percentage compared to using only the linear penalty term.

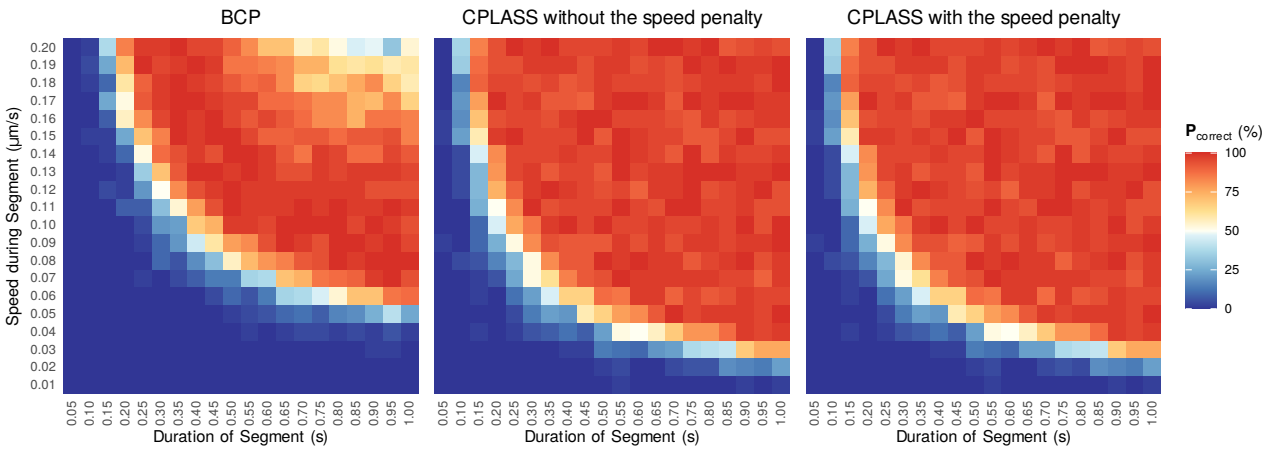


Figure 8. *Power analysis comparing CPLASS and BCP (Section 3.2).* A total of 20,000 trajectories were simulated at 20 Hz with two true changepoints ($\sigma = 0.01$). The first and third segments have fixed durations of 2 s, while the middle segment varies across 20 durations and 20 speeds (μm/s). For each speed–duration pair, 50 trajectories were generated. The detection rate $\mathbf{P}_{\text{correct}}$ is the percentage of paths for which the correct number of changepoints is identified. Trajectories follow the same simulation design as Figure 6(C),(D).

3.3. Comparison with alternative changepoint methods

We compared CPLASS version 0.1.1 with CPOP version 1.0.9 [12], NOT version 1.6 [16], first-order trend filtering implemented in `genlasso` version 1.6.1 [14], PLANT [24], and the diffusion–velocity changepoint method of [23]. PLANT is formulated for scalar time traces and was applied by [24] to coordinate traces obtained from cell-tracking experiments; it does not impose a joint common changepoint segmentation across spatial coordinates. In contrast, the framework of [23] includes an explicit multidimensional extension under a drift–diffusion model.

We used the publicly available implementations of CPOP, NOT, trend filtering, PLANT, and the diffusion–velocity method. Repository commits and method-specific settings are reported in Supplementary Section S.6.

All methods were evaluated using a common frozen collection of two-dimensional trajectories. Observations were recorded at $t_i = (i-1)\Delta, i = 1, \ldots, 251$, with sampling interval $\Delta = 0.04$ s, corresponding to 25 Hz over 10 s. Following the model defined in Section 2.1, we generated

$$Y_i = f^0_{\boldsymbol{\tau},\mathbf{V},\underline{a}}(t_i) + \varepsilon_i, \qquad \varepsilon_i \overset{\text{iid}}{\sim} N_2\left(0, \sigma_0^2 \mathbf{I}_2\right),$$

where $Y_i = (x_i, y_i)^{\mathsf{T}} \in \mathbb{R}^2$ and $f^0_{\boldsymbol{\tau},\mathbf{V},\underline{a}}$ is the continuous piecewise-linear function determined by the changepoint vector $\boldsymbol{\tau}$, segment-velocity parameters $\mathbf{V}$, and intercept $\underline{a}$.

Under the alternative, the changepoint vector was $\boldsymbol{\tau} = (2, 5, 7)$ s, producing four segments of durations 2, 3, 2, and 3 s. The corresponding segment velocities are given in Table 1. Under the matched null model, $\boldsymbol{\tau} = \varnothing$ and the latent trajectory had constant velocity $\mathbf{V}_0$.

We generated 100 independent alternative trajectories and 100 independent null trajectories for each regime. The trajectories were generated once and then held fixed so that every method was evaluated

Table 1. Generating parameters for the method-comparison simulations. Each ordered pair denotes (v_x, v_y) in μm/s.

Regime	σ_0	$\mathbf{V}_1$	$\mathbf{V}_2$	$\mathbf{V}_3$	$\mathbf{V}_4$	Null velocity $\mathbf{V}_0$
Clear changes	0.015	$(0, 0)$	$(0.35, 0.10)$	$(0, 0)$	$(-0.15, 0.30)$	$(0.060, 0.120)$
Weak joint changes	0.030	$(0.08, 0.08)$	$(0.15, 0.02)$	$(0.07, -0.09)$	$(-0.02, -0.15)$	$(0.069, -0.041)$
Noisy pauses	0.050	$(0, 0)$	$(0.25, 0.12)$	$(0, 0)$	$(-0.20, 0.18)$	$(0.015, 0.090)$

using exactly the same observations.

CPLASS was applied jointly to the two spatial coordinates. Methods designed for scalar observations were applied separately to the x- and y-coordinates, and their reported changepoint sets were combined by exact union: $\widehat{\mathcal{M}} = \widehat{\mathcal{M}}_x \cup \widehat{\mathcal{M}}_y$. Although the framework of [23] includes a multidimensional extension, the publicly released executable workflow that we were able to reproduce analyzed one coordinate at a time. We therefore evaluated that implementation on both coordinates and combined its detections by the same exact-union rule. We did not merge nearby coordinate-wise detections because doing so would introduce an additional post-processing rule and tuning parameter not specified by the released implementations.

For the alternative simulations, let $\mathcal{M} = \{51, 126, 176\}$ denote the true changepoint indices, and let $\widehat{\mathcal{M}}$ denote the inferred set. We constructed a one-to-one matching between the true and inferred changepoints, allowing a match only when $|\widehat{\tau} - \tau| \leq 5$ observations, corresponding to 0.20 s. This window was small relative to the shortest true segment duration of 2 s. If M denotes the number of matched pairs, $K = |\mathcal{M}| = 3$, and $\widehat{K} = |\widehat{\mathcal{M}}|$, performance was summarized by

$$\text{Precision} = \frac{M}{\widehat{K}}, \qquad \text{Recall} = \frac{M}{K}, \qquad \text{Count ratio} = \frac{\widehat{K}}{K}.$$

Precision was defined as zero when $\widehat{K} = 0$. Under the null, Type I error was defined as

$$\Pr(\widehat{K} > 0 \mid H_0),$$

the probability of selecting at least one changepoint when none was present.

CPLASS achieved mean precision of 93.0%–99.8% and mean recall of 93.0%–100% across the three regimes (Figure 9(A),(B)). Its mean inferred-to-true changepoint-count ratio remained between 1.00 and 1.01, indicating little systematic over- or undersegmentation (Figure 9(C)). CPOP and NOT attained recall of 95.7%–100%, but their precision was 49.6%–60.2%, reflecting mean changepoint-count ratios of approximately 1.70–1.97. Trend filtering showed a similar distinction between recovery and segmentation accuracy: recall was 95.7%–100%, but the inferred count was approximately 3.53–4.16 times the true count. PLANT attained 100% mean recall in the clear-change regime but had mean recall of 26.3% and 32.7% in the weak-joint and noisy-pause regimes, respectively.

Under the null, CPLASS selected at least one changepoint in 0%–2% of trajectories, compared with 1%–8% for CPOP and NOT, 0% for trend filtering, and 15%–22% for PLANT (Figure 9(D)). The implementation of Yin et al. produced extensive oversegmentation and selected changepoints in every null trajectory. This result should be interpreted cautiously because that method was developed for a different drift–diffusion observation model, whereas the present benchmark contains independent positional localization errors. Its inclusion therefore evaluates behavior under the observation model relevant to CPLASS rather than performance under its native assumptions.

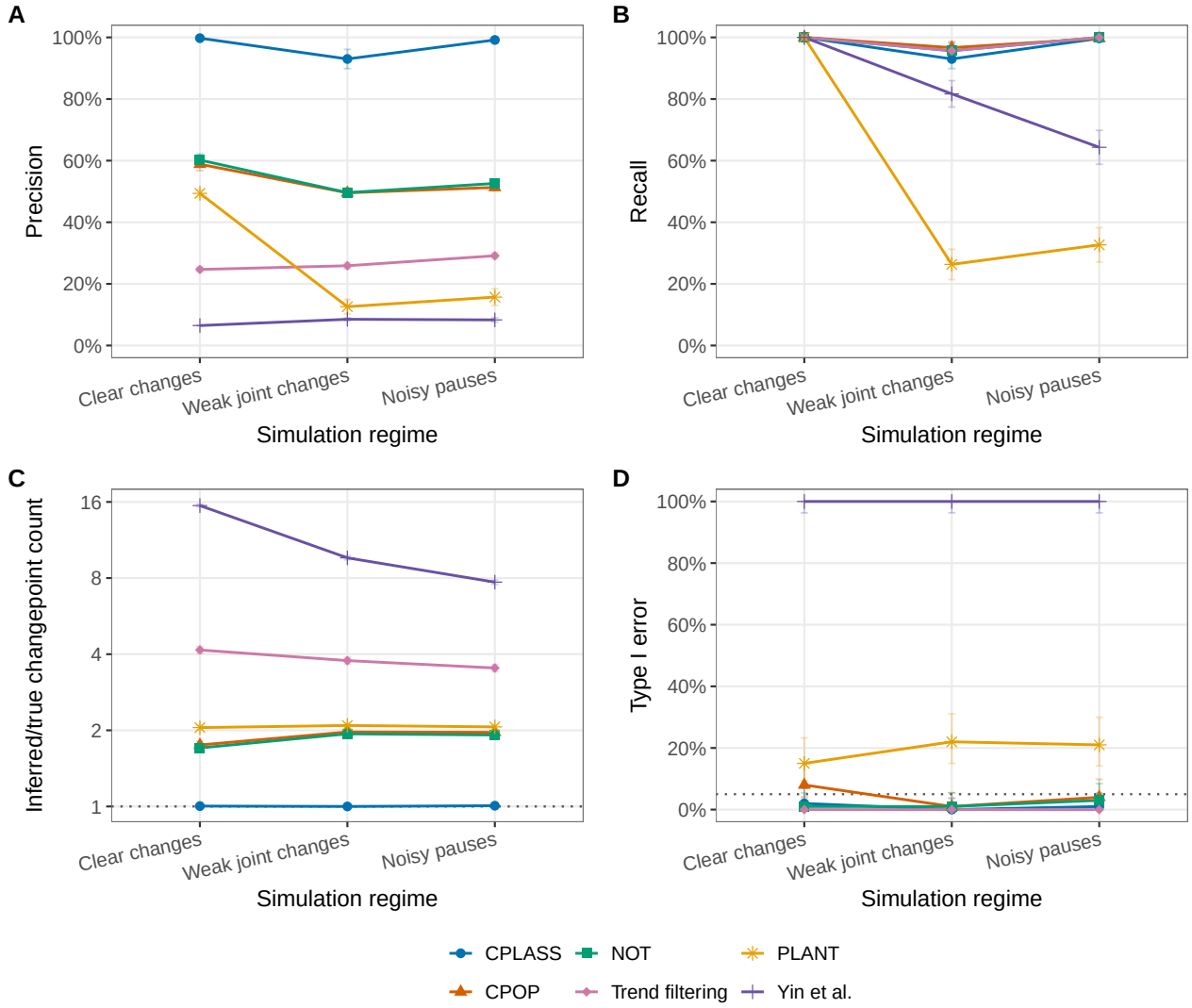


Figure 9. *Comparison of CPLASS with alternative changepoint methods.* Results are based on 100 frozen trajectories per method and simulation regime. **(A)** Precision, defined as the proportion of inferred changepoints matched to a generating changepoint. **(B)** Recall, defined as the proportion of generating changepoints successfully localized. In panels A and B, matches were determined one-to-one within five observations (0.20 s). **(C)** Ratio of the inferred to the true number of changepoints. The dotted line denotes the correct ratio of one, and the vertical axis is logarithmic. **(D)** Type I error, defined as the proportion of null trajectories for which at least one changepoint was selected. The dotted line denotes 5%. Scalar methods were applied separately to the x- and y-coordinates, followed by exact union of the reported changepoints. Error bars in panels A–C show normal-approximation 95% intervals for the trajectory-level means; error bars in panel D show 95% Wilson intervals.

3.4. *In vivo experimental data—lysosomal transport*

In this section, we revisit the datasets in [25]. Two cell lines, monkey kidney epithelial cells, and human lung epithelial cells were cultured in different media but with identical conditions. Cells were supplemented with fetal bovine serum and incubated at 37 °C and 5% carbon dioxide. For imaging experiments, cells were transduced with CellLight Lysosomes-green emerald fluorescent protein to label lysosomes fluorescently. Transduction was carried out according to the manufacturer's instructions. The study used live cell imaging and single-particle tracking to observe and characterize lysosome motion. A confocal microscope was used to collect images, and the TrackMate [37] macro was used to track lysosomes. Lysosome trajectories were labeled as being in the perinuclear or peripheral regions of the cells and were sorted by size. We refer to [25] for more details about the datasets and data processing.

This application is a methodological reanalysis of the previously published experiment rather than an independent biological replication. The tracked trajectories are the observational units analyzed by CPLASS, whereas the cell preparations, imaging experiments, and biological replication are those of the original study [25]. We therefore interpret the following comparisons descriptively and do not treat the number of trajectories as the number of independent biological replicates. The new contribution is the joint continuous two-dimensional segmentation and the threshold-free CSA summary of the full allocation of tracked time across inferred speeds.

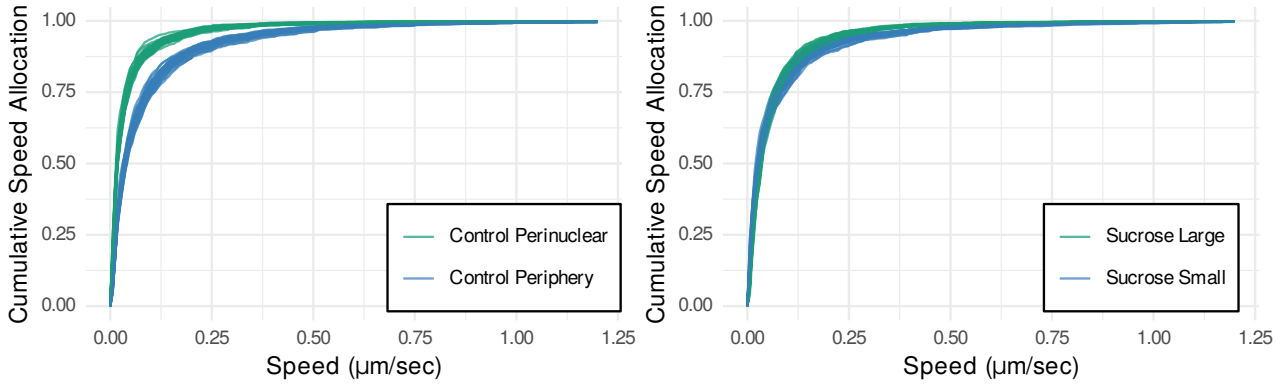


Figure 10. *CSA plots for different group comparisons (Section 3.4).* (**Left**) The CSA bootstrap ensemble curves compare lysosomal transport in perinuclear and periphery regions. (**Right**) The CSA bootstrap ensemble curves compare the sucrose-treated groups restricted to the periphery region of the cell from [25].

We reassessed the queries regarding how lysosomal transport varies with lysosome size and location. Figure 10 indicates that intracellular location, rather than diameter, is a crucial factor in lysosomal motion. This aligns with the findings from [25]. Analyzing the CSA plot (the left panel of Figure 10), we observe that the lysosome in the perinuclear region spends more time moving slowly compared to that in the peripheral region. The right panel of Figure 10 shows a more modest separation between large and small lysosomes than the separation between peripheral and perinuclear trajectories. Thus, intracellular location is the more prominent of the two recorded grouping variables in these descriptive CSA comparisons, while large lysosomes still tend to allocate more time to lower speeds than small lysosomes. In [25], to study the differences between these group comparisons, the authors first classify

segment speeds into groups of motile ($s > 0.1$ μm/s) and stationary ($s < 0.1$ μm/s), then analyze the empirical cumulative distribution for the motile group. Meanwhile, using CPLASS and CSA, we can analyze the entire collection of segmented speeds and durations without establishing a threshold for the motile segment group.

3.5. In vitro experimental data—quantum dot transport

This section revisited the datasets used in [22] in which quantum dots are transported by a single kinesin-1 (kin-1) motor, a single dynein-dynactin-BicD2 (DDB) motor, and by a kinesin-1/DDB pair. In [22], the authors developed a protocol for finding changepoints in cargo trajectories that can be projected along the length of a straight microtubule and reporting velocity distributions. The differences in velocity distributions and run lengths revealed the behavioral difference for the biophysical state of three distinct motor-cargo arrangements. Since our proposed CPLASS can handle multidimensional datasets, we applied it directly to these quantum dot datasets without projecting the two-dimensional data into a one-dimensional format. We then calculated the CSA bootstrap ensemble curves (see the left panel of Figure 11) based on the collection of estimated segment speeds and durations in each motor-cargo system after running CPLASS. The CSA plot illustrates the differences among the motor experiments that correspond with what one might expect in a molecular motor "folklore." In other words, kinesin-1 motors step processively with consistent behavior, while DDB motors (orange curves) exhibit a broader range of speeds. When both motor types are present (green curves), the speed is generally lower, possibly reflecting a tug-of-war state. The right panel of Figure 11—weighted kernel density estimation (WKDE) of inferred segment speeds with weights given by segment durations across the three groups—is a good tool for visualizing our argument. The tug-of-war phenomenon is more evident in this plot with more slow segments: One mode is near 0 (Kin1DDB), and the second mode is evident for cases when it appears one motor is dominant. The multimodality we observe in the WKDE here is also reported in the paper of [22]. All of these confirm the observations made about the datasets but offer a more refined and robust characterization.

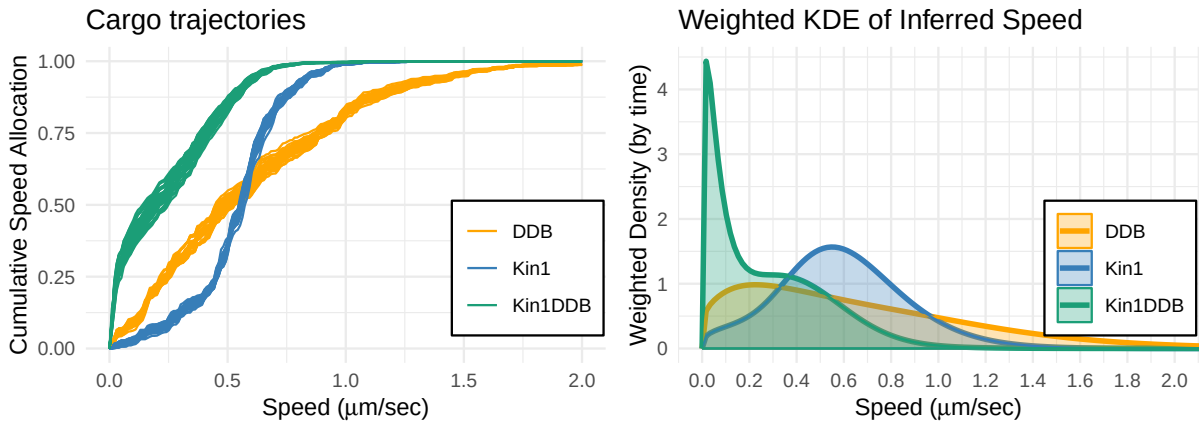


Figure 11. *CSA plot for different motor families (Section 3.5).* (**Left**) The CSA bootstrap ensemble curves compare cargo trajectories among three groups of motor transports: kinesin-1 (Kin1), dynein-dynactin-BicD2 (DDB), and kinesin-1/DDB pairs. (**Right**) WKDE of inferred segment speeds, duration-weighted, for the three groups.

3.6. Preliminary application to microtubule dynamics in neurons

Microtubules constantly undergo cycles of polymerization and depolymerization within cells, providing both mechanical stability and transport infrastructure [38]. In neurons, microtubule polarity helps organize intracellular transport, and the growing plus ends of microtubules can be labeled with end-binding proteins such as EB1-GFP, producing fluorescent "comets" whose trajectories provide a live-cell readout of microtubule polymerization dynamics [27]. As a preliminary application, we applied CPLASS to EB1-GFP comet trajectories recorded in class I dendritic arborization (da) sensory neurons in *Drosophila* larvae.

The data analyzed here represent two videos, denoted top 2 and top 3, from a larger collection of 20 imaging sessions of the same cell type. The labels top 2 and top 3 refer to the orientation of the imaged dendritic field relative to the neuronal cell body, rather than to distinct experimental conditions. Full analysis of the complete dataset is ongoing. Comet tracking from raw fluorescence video requires substantial manual curation [27]. TrackMate identified 47 candidate comet trajectories across the two videos; after excluding 17 short paths below the minimum observation count required for stable CPLASS fitting, 30 trajectories remained for analysis (top 2: 14 trajectories, split across left and right sub-regions of the imaged dendrite; top 3: 16 trajectories). The imaging field of view was $34.97 \times 34.97\,\mu$m with a time increment of 1.79 s between frames.

Each trajectory was segmented by CPLASS into a sequence of piecewise-linear segments, producing estimates of local comet speed and segment duration. The number of inferred segments varied across tracks, reflecting heterogeneity in comet motion along different dendritic branches. In total, the analysis produced 106 inferred segments over 1348 s of tracked duration for top 2 and 84 inferred segments over 1341 s for top 3. The central 90% of tracked duration corresponded to inferred segment speeds of approximately 0.029–0.185 μm/s (1.73–11.1 μm/min) for top 2 and 0.021–0.171 μm/s (1.25–10.3 μm/min) for top 3. The duration-weighted mean speeds were 6.32 μm/min for top 2 and 5.92 μm/min for top 3. These values are broadly consistent with previously reported EB1 comet speed and run-length measurements in neuronal dendrites [39].

Figure 12 summarizes the CPLASS output across both animals using CSA and the duration-weighted kernel density estimate (WKDE) of inferred segment speeds, both reported in μm/min. The WKDE shows a unimodal, right-skewed distribution for both animals, with duration-weighted mean speeds of 6.32 μm/min for top 2 and 5.92 μm/min for top 3. The CSA curves show similar allocation patterns over the main speed range, indicating that the two preliminary datasets provide comparable summaries of EB1-GFP comet motion. Occasional faster inferred segments are present, but they contribute relatively little to the duration-weighted distribution.

As in our lysosomal and quantum dot analyses (Sections 3.4 and 3.5), the CSA summarizes the full distribution of inferred segment speeds without requiring an arbitrary speed threshold to classify segments. In this setting, the segment-level output from CPLASS provides a compact way to summarize local variation in EB1-GFP comet motion along dendritic trajectories. In future work, once the full dataset has been processed, these summaries could be used to compare polymerization behavior across animals, dendritic regions, comet orientations, or genotypes.

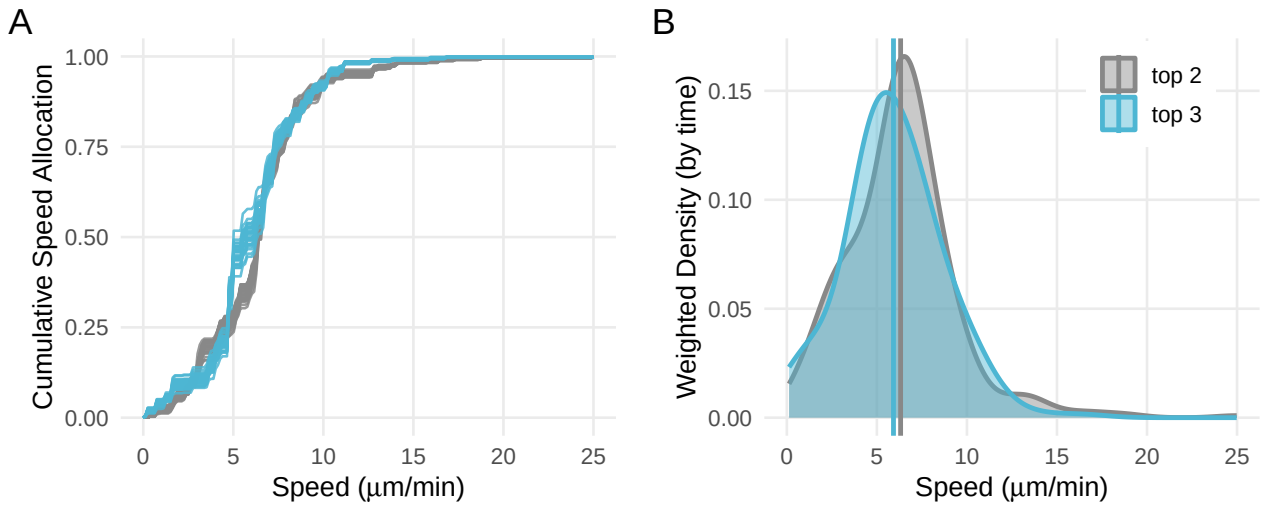


Figure 12. *CSA and WKDE plots for EB1-GFP microtubule comet trajectories (Section 3.6).* **(A)** CSA bootstrap ensemble curves compare inferred speeds across two larvae of the same cell type (top 2: 14 trajectories, gray; top 3: 16 trajectories, teal). **(B)** Duration-weighted kernel density estimates (WKDEs) of inferred segment speeds for the same two animals. Vertical lines indicate the duration-weighted mean speed for each animal (top 2: 6.32 μm/min, gray; top 3: 5.92 μm/min, teal). Both panels report speeds in μm/min. The similar distributions across the two animals suggest that CPLASS provides a stable summary of EB1-GFP comet speed in this preliminary dataset.

We note one main limitation of the current analysis. CPLASS is applied here in its standard piecewise-linear form, which does not yet explicitly account for the curved and branched geometry of neuronal dendrites. A generalization to non-linear piecewise paths suited to spatially complex neuronal geometries is in active development.

Despite this limitation, these results demonstrate that CPLASS produces biologically plausible segmentations of EB1-GFP comet trajectories without requiring manual annotation of individual segments. The CSA statistic provides a stable preliminary summary for comparing comet-speed distributions across individuals and, in future work, across broader biological conditions.

3.7. Additional robustness analyses

We conducted two additional sensitivity analyses addressing departures from the Gaussian observation-error model. First, we replaced Gaussian coordinate errors with compactly supported uniform errors having the same variance. Second, we examined the effect of adding a single isolated positional outlier. Together, these experiments comprised 300 independently generated trajectories and 600 CPLASS fits, including analyses with and without the speed penalty. The complete generative models, parameter settings, evaluation criteria, numerical results, and representative fits are provided in Supplementary Section S.4.

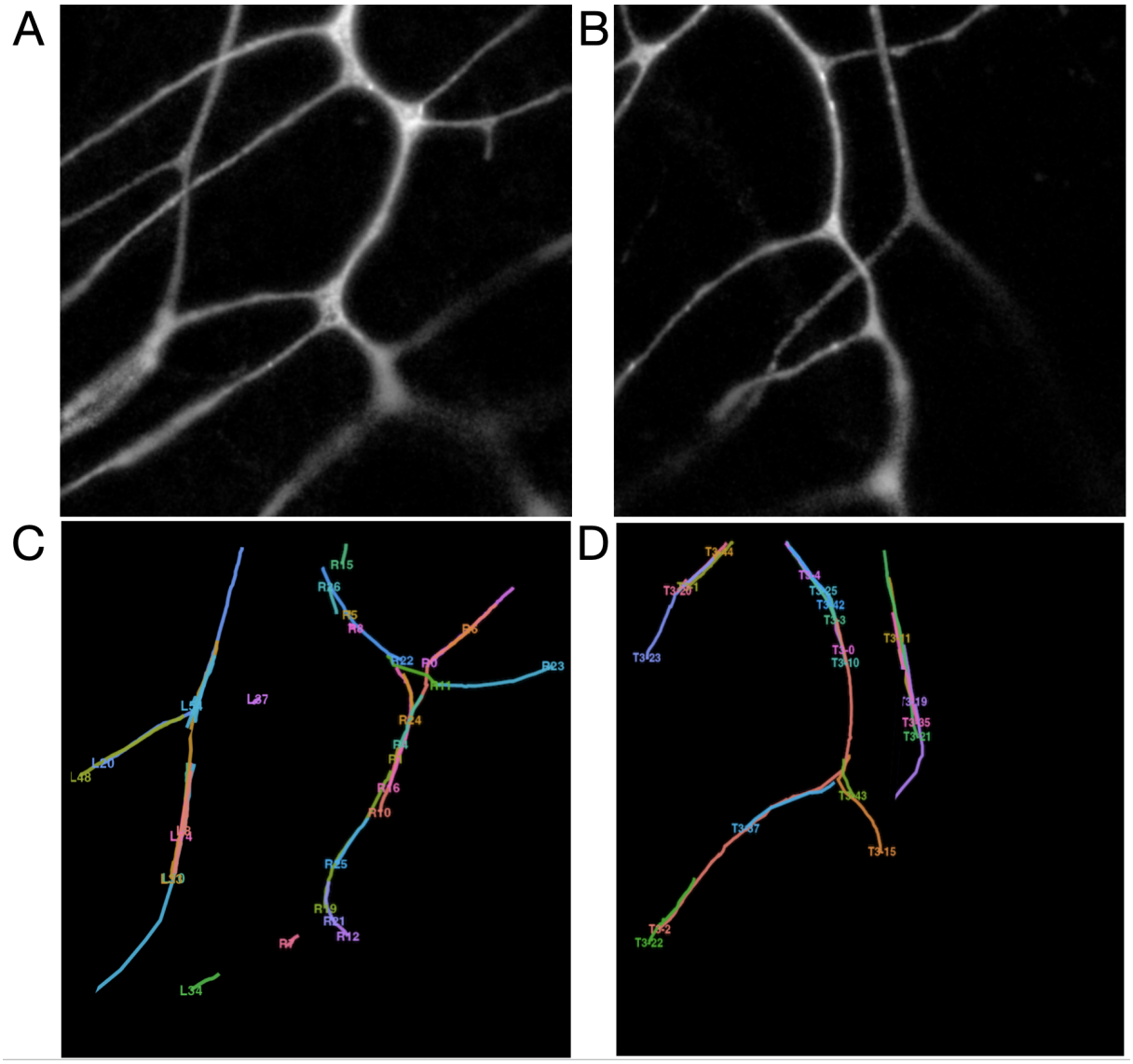


Figure 13. *EB1-GFP microtubule comet imaging and TrackMate trajectory output (Section 3.6).* (**A,B**) Representative first frames from two imaging videos of the same cell type (class I da sensory neuron) from two different larvae of the same genotype (top 2 and top 3, respectively). Each video is a zoomed-in field of view of the dendritic arbor; the neuron's cell body (soma) lies outside the imaged region, above the field shown, and dendrite direction is defined relative to it ("top" denotes the side nearer the soma, used to establish comet travel direction). EB1-GFP labels the growing plus ends of microtubules within these dendrites. The bright branching structures illustrate the curved and spatially complex geometry along which comets were tracked. (**C**) Comet trajectories extracted from the top 2 video using TrackMate, with each color representing a distinct tracked comet. (**D**) Comet trajectories for the top 3 video. These trajectories serve as input to CPLASS, which segments each path into piecewise-linear velocity segments as summarized in Figure 12. The branched and curved trajectory shapes in panels C and D reflect the underlying dendritic geometry visible in panels A and B.

4. Discussion

In this work, we introduced the CPLASS algorithm for detecting changes in velocity within multidimensional data, addressing key challenges in both probability structure and search methodology. While detecting changes in velocity seems to be a similar statistical problem to detecting changes in mean, it is fundamentally more challenging. Popular generic approaches for detecting multiple changepoints are not directly applicable to continuous change-in-velocity models. For example, popular recursive strategies, like binary segmentation, are initiated by assuming a single changepoint and finding the most likely location. However, as we discuss in the main text, the changepoint time inferred under the assumption that there is one change may not appear among (or near) the changepoints detected assuming there are two. This is a marked difference from the change-in-mean detection problem. Moreover, standard dynamic programming algorithms like PELT and optimal partitioning cannot directly handle continuous change-in-slope models because continuity creates dependencies between the parameters of neighboring segments, violating the segment-additivity structure required by these algorithms. To address related issues, [15] proposed the NOT algorithm, [12] introduced a specialized dynamic-programming method, and [14] developed trend-filtering approaches. PLANT [24] reconstructs scalar noisy traces using adjacent linear segments, although continuity between neighboring fitted segments is not imposed. In a distinct drift–diffusion framework, [23] developed a likelihood-ratio and recursive-segmentation procedure for detecting changes in velocity, diffusion, or both, including an extension to multidimensional trajectories. These methods provide important solutions to related changepoint problems, but they do not jointly combine a globally continuous piecewise-linear latent-position model, a common segmentation across spatial coordinates, and simultaneous selection of the complete changepoint configuration. These challenges motivated our development of an MCMC-based approach, which includes specialized proposal mechanisms tailored to efficiently navigate the parameter space.

While consistency of the CPLASS estimator is established formally in our companion theoretical work [34], real-world applications, such as molecular motor data, pose additional challenges because practitioners often have prior knowledge of a practical upper bound on attainable speeds. To incorporate this information, we introduced the dimensionless speed regularizer $\eta \sum_j \left(\frac{\widehat{s}_j}{s_{\text{cap}}} - 1 \right)_+$, where s_{cap} is a physically specified reference speed and η controls the strength of the penalty. This regularizer discourages unrealistically short segments that imply implausibly high fitted speeds, without penalizing segments whose speeds remain below s_{cap}. Our sensitivity analysis showed that increasing η progressively reduced extreme fitted speeds in trajectories for which the penalty was active, while leaving low-speed simulations essentially unchanged. The formal consistency result applies to the sSIC-only criterion, recovered by setting $\eta = 0$; the optional speed term is presented separately as a finite-sample, domain-informed regularizer and is not included in the claimed consistency theorem. Furthermore, by comparing CSA [29] with the CDF, we showed that quantifying the proportion of time spent at different speeds yields a more stable performance metric than segment velocity counts, making the approach less sensitive to algorithmic variation.

The comparison with alternative changepoint methods further illustrates the practical consequences of these modeling choices. Across the three simulation regimes, CPLASS maintained high precision and recall, selected approximately the correct number of changepoints, and controlled false detections under the null. Several competing methods recovered most true changepoints but also produced additional detections, particularly when coordinate-wise changepoint sets were combined. These results support

jointly estimating a common segmentation across coordinates, while recognizing that some competing methods were developed under observation models different from the positional-error model considered here.

Crucially, CPLASS is formulated for multidimensional trajectories and estimates a common changepoint segmentation jointly across spatial coordinates. The matrix formulation extends directly to dimensions greater than two, and the accompanying theoretical results are stated for the multidimensional setting. The current R package and empirical evaluations focus on two-dimensional trajectories because they were developed for the intracellular-transport applications considered here; computational scaling with increasing dimension and a general higher-dimensional software interface remain topics for future work. Ensuring the consistency of CSA inference following segmentation remains an open theoretical question that we intend to address in future work. Improvements in functional inference could have an immediate impact on modern single-particle-tracking investigations in intracellular biophysics.

The preliminary application to EB1-GFP microtubule plus-end trajectories in neurons (Section 3.6) highlights a third direction for future work: extending CPLASS to nonlinear piecewise paths suited to the curved and branched geometry of neuronal dendrites. In its current piecewise-linear form, CPLASS produces biologically plausible and reproducible segmentations of microtubule polymerization dynamics, with inferred speeds consistent with known growth rates in neuronal dendrites. A generalization to nonlinear paths would enable direct association of local polymerization speed with cellular geometry, connecting our statistical framework with emerging theoretical models of microtubule organization [28, 40].

CPLASS nevertheless has several limitations. The current formulation assumes a continuous piecewise-linear latent trajectory with independent Gaussian positional errors, and it does not include an explicit component for isolated measurement artifacts. Our supplementary sensitivity analysis showed that changepoint recovery was maintained under variance-matched, compactly supported errors, but an isolated positional outlier was occasionally represented as a short regime bounded by two nearby changepoints. Consequently, tracking-quality screening or a robust observation model may be needed when such artifacts are common. The stochastic search also requires multiple starts to reduce sensitivity to initialization, increasing computation relative to a single-chain analysis. Future extensions could therefore incorporate robust observation losses and more efficient search strategies.

Use of AI tools declaration

During the preparation of this work, the author used ChatGPT and Claude to assist with debugging and computational optimization of the R package implementation. The original CPLASS code was written by the author and is available in the original repository. All AI-assisted outputs were reviewed and verified by the author, who takes full responsibility for the code and the content of this article.

Data and code

The `cplass` R package implementing the CPLASS algorithm is publicly available at `https://github.com/linhdo154/cplass`. The repository includes source code, documentation, tests, illustrative examples, and installation instructions. A companion documentation website is available at `https://linhdo154.github.io/cplass/`.

Acknowledgments

This research was supported by the NSF-Simons Southeast Center for Mathematics and Biology (SCMB) through the grants National Science Foundation DMS1764406 and Simons Foundation/SFARI 594594. This work was also supported by National Institutes of Health grant R01NS121245. We thank Christine K. Payne, William O. Hancock, and Melissa M. Rolls for many substantive discussions and sharing insights and experimental data results from their labs.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. S. P. Gross, Hither and yon: a review of bi-directional microtubule-based transport, *Phys. Biol.*, **1** (2004), R1. https://doi.org/10.1088/1478-3967/1/2/R01
2. A. Kunwar, S. K. Tripathy, J. Xu, M. K. Mattson, P. Anand, R. Sigua, et al., Mechanical stochastic tug-of-war models cannot explain bidirectional lipid-droplet transport, *PNAS*, **108** (2011), 18960–18965. https://doi.org/10.1073/pnas.1107841108
3. W. O. Hancock, Bidirectional cargo transport: Moving beyond tug of war, *Nat. Rev. Mol. Cell Biol.*, **15** (2014), 615–628. https://doi.org/10.1038/nrm3853
4. M. Lavielle, G. Teyssière, Adaptive detection of multiple change-points in asset price volatility, in *Long Memory in Economics*, Springer Berlin Heidelberg, Berlin, Heidelberg, (2007), 129–156. https://doi.org/10.1007/978-3-540-34625-8_5
5. J. Bai, P. Perron, estimating and testing linear models with multiple structural changes, *Econometrica*, **66** (1998), 47–78. https://doi.org/10.2307/2998540
6. K. Frick, A. Munk, H. Sieling, Multiscale change point inference, *J. R. Stat. Soc. B*, **76** (2014), 495–580. https://doi.org/10.1111/rssb.12047
7. A. Futschik, T. Hotz, A. Munk, H. Sieling, Multiscale DNA partitioning: statistical evidence for segments, *Bioinformatics*, **30** (2014), 2255–2262. https://doi.org/10.1093/bioinformatics/btu180
8. Y. Niu, H. Zhang, The screening and ranking algorithm to detect DNA copy number variations, *Ann. Appl. Stat.*, **6** (2012), 1306–1326. https://doi.org/10.1214/12-AOAS539
9. R. Killick, I. A. Eckley, K. Ewans, P. Jonathan, Detection of changes in variance of oceanographic time-series using changepoint analysis, *Ocean Eng.*, **37** (2010), 1120–1126. https://doi.org/10.1016/j.oceaneng.2010.04.009
10. R. Killick, P. Fearnhead, I. A. Eckley, Optimal detection of changepoints with a linear computational cost, *J. Am. Stat. Assoc.*, **107** (2012), 1590–1598. https://doi.org/10.1080/01621459.2012.737745
11. B. Jackson, J. Scargle, D. Barnes, S. Arabhi, A. Alt, P. Gioumousis, et al., An algorithm for optimal partitioning of data on an interval, *IEEE Signal Process. Lett.*, **12** (2005), 105–108. https://doi.org/10.1109/LSP.2001.838216

12. P. Fearnhead, R. Maidstone, A. Letchford, Detecting changes in slope with an ℓ_0 penalty, *J. Comput. Graphical Stat.*, **28** (2019), 265–275. https://doi.org/10.1080/10618600.2018.1512868

13. P. Fearnhead, D. Grose, cpop: Detecting changes in piecewise-linear signals, *J. Stat. Software*, **109** (2024), 1–30. https://doi.org/10.18637/jss.v109.i07

14. S. J. Kim, K. Koh, S. Boyd, D. Gorinevsky, ℓ_1 trend filtering, *SIAM Rev.*, **51** (2009), 339–360. https://doi.org/10.1137/070690274

15. R. Baranowski, Y. Chen, P. Fryzlewicz, Narrowest-over-threshold detection of multiple change points and change-point-like features, *J. R. Stat. Soc. B*, **81** (2019), 649–672. https://doi.org/10.1111/rssb.12322

16. P. Fryzlewicz, Narrowest significance pursuit: inference for multiple change-points in linear models, *J. Am. Stat. Assoc.*, **119** (2024), 1633–1646. https://doi.org/10.1080/01621459.2023.2211733

17. N. Monnier, S. M. Guo, M. Mori, J. He, P. Lénárt, M. Bathe, Bayesian approach to MSD-based analysis of particle motion in live cells, *Biophys. J.*, **103** (2012), 616–626. https://doi.org/10.1016/j.bpj.2012.06.029

18. C. Manzo, M. F. Garcia-Parajo, A Review of progress in single particle tracking: From methods to biophysical insights, *Rep. Prog. Phys.*, **78** (2015), 124601. https://doi.org/10.1088/0034-4885/78/12/124601

19. F. Persson, M. Lindén, C. Unoson, J. Elf, Extracting intracellular diffusive states and transition rates from single-molecule tracking data, *Nat. Methods*, **10** (2013), 265–269. https://doi.org/10.1038/nmeth.2367

20. J. D. Karslake, E. D. Donarski, S. A. Shelby, L. M. Demey, V. J. DiRita, S. L. Veatch, et al., SMAUG: Analyzing single-molecule tracks with nonparametric Bayesian statistics, *Methods*, **193** (2021), 16–26. https://doi.org/10.1016/j.ymeth.2020.03.008

21. S. Neumann, R. Chassefeyre, G. E. Campbell, S. E. Encalada, KymoAnalyzer: A software tool for the quantitative analysis of intracellular transport in neurons, *Traffic*, **18** (2017), 71–88. https://doi.org/10.1111/tra.12456

22. M. A. Jensen, Q. Feng, W. O. Hancock, S. A. McKinley, A change point analysis protocol for comparing intracellular transport by different molecular motor combinations, *Math. Biosci. Eng.*, **18** (2021), 8962–8996. https://doi.org/10.3934/mbe.2021442

23. S. Yin, N. Song, H. Yang, Detection of velocity and diffusion coefficient change points in single-particle trajectories, *Biophy. J.*, **115** (2018), 217–229. https://doi.org/10.1016/j.bpj.2017.11.008

24. A. Sosa-Costa, I. K. Piechocka, L. Gardini, F. S. Pavone, M. Capitanio, M. F. Garcia-Parajo, et al., PLANT: A method for detecting changes of slope in noisy trajectories, *Biophys. J.*, **114** (2018), 2044–2051. https://doi.org/10.1016/j.bpj.2018.04.006

25. N. T. Rayens, K. J. Cook, S. A. McKinley, C. K. Payne, Transport of lysosomes decreases in the perinuclear region: Insights from changepoint analysis, *Biophys. J.*, **121** (2022), 1205–1218. https://doi.org/10.1016/j.bpj.2022.02.032

26. P. Fryzlewicz, S. Rao, Multiple-change-point detection for auto-regressive conditional heteroscedastic processes, *J. R. Stat. Soc. B*, **76** (2014), 903–924. https://doi.org/10.1111/rssb.12054

27. P. Thyagarajan, C. Feng, D. Lee, M. Shorey, M. M. Rolls, Microtubule polarity is instructive for many aspects of neuronal polarity, *Dev. Biol.*, **486** (2022), 56–70. https://doi.org/10.1016/j.ydbio.2022.03.009

28. H. G. Scanlon, G. Mahata, A. C. Nelson, S. A. McKinley, M. M. Rolls, M. V. Ciocanel, Nucleation feedback can drive establishment and maintenance of biased microtubule polarity in neurites, *Math. Biosci.*, **389** (2025), 109538. https://doi.org/10.1016/j.mbs.2025.109538

29. K. J. Cook, N. Rayens, L. Do, C. K. Payne, S. A. McKinley, Considering experimental frame rates and robust segmentation analysis of piecewise-linear microparticle trajectories, *Math. Biosci. Eng.*, **22** (2025), 2595–2626. https://doi.org/10.3934/mbe.2025095

30. M. Lavielle, Optimal segmentation of random processes, *IEEE Trans. Signal Process.*, **46** (1998), 1365–1373. https://doi.org/10.1109/78.668798

31. P. Fryzlewicz, Wild binary segmentation for multiple change-point detection, *Ann. Stat.*, **42** (2014), 2243 – 2281. https://doi.org/10.1214/14-AOS1245

32. Y. C. Yao, Estimating the number of change-points via Schwarz' criterion, *Stat. Probab. Lett.*, **6** (1988), 181–189. https://doi.org/10.1016/0167-7152(88)90118-6

33. T. L. Do, *Changepoint Detection: Theory, Algorithms, and Application in Intracellular Transport*, Ph.D dissertation, Tulane University, 2025,

34. L. Do, D. Do, S. A. McKinley, Consistency of penalized maximum likelihood estimation for multidimensional change-in-velocity detection, preprint, arXiv:2610.04191, 2026. https://arxiv.org/abs/2610.04191

35. M. Lavielle, E. Lebarbier, An application of MCMC methods for the multiple change-points problem, *Signal Process.*, **81** (2001), 39–53. https://doi.org/10.1016/S0165-1684(00)00189-4

36. P. J. Green, Reversible jump Markov chain Monte Carlo computation and Bayesian model determination, *Biometrika*, **82** (1995), 711–732. https://doi.org/10.1093/biomet/82.4.711

37. J. Y. Tinevez, N. Perry, J. Schindelin, G. M. Hoopes, G. D. Reynolds, E. Laplantine, et al., TrackMate: An open and extensible platform for single-particle tracking, *Methods*, **115** (2017), 80–90. https://doi.org/10.1016/j.ymeth.2016.09.016

38. M. K. Gardner, M. Zanic, J. Howard, Microtubule catastrophe and rescue, *Curr. Opin. Cell Biol.*, **25** (2013), 14–22. https://doi.org/10.1016/j.ceb.2012.09.006

39. C. Feng, P. Thyagarajan, M. Shorey, D. Y. Seebold, A. T. Weiner, R. M. Albertson, et al., Patronin-mediated minus end growth is required for dendritic microtubule polarity, *J. Cell Biol.*, **218** (2019), 2309–2328. https://doi.org/10.1083/jcb.201810155

40. A. C. Nelson, M. M. Rolls, M. V. Ciocanel, S. A. McKinley, Minimal mechanisms of microtubule length regulation in living cells, *Bull. Math. Biol.*, **86** (2024), 58. https://doi.org/10.1007/s11538-024-01279-z



AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)