Biophysical Journal
Article

Biophysical Society

Challenges in Estimating the Motility Parameters of Single Processive Motor Proteins

Felix Ruhnow,[1] Linda Kloβ,[2] and Stefan Diez[1,2,*]
[1]B CUBE - Center for Molecular Bioengineering, Technische Universität Dresden, Dresden, Germany and [2]Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany

ABSTRACT Cytoskeletal motor proteins are essential to the function of a wide range of intracellular mechano-systems. The biophysical characterization of their movement along their filamentous tracks is therefore of large importance. Toward this end, single-molecule, in vitro stepping-motility assays are commonly used to determine motor velocity and run length. However, comparing results from such experiments has proved difficult due to influences from variations in the experimental conditions and the data analysis methods. Here, we investigate the movement of fluorescently labeled, processive, dimeric motor proteins and propose a unified algorithm to correct the measurements for finite filament length as well as photobleaching. Particular emphasis is put on estimating the statistical errors associated with the proposed evaluation method, as knowledge of these values is crucial when comparing measurements from different experiments. Testing our approach with simulated and experimental data from GFP-labeled kinesin-1 motors stepping along immobilized microtubules, we show 1) that velocity distributions should be fitted by a *t* location-scale probability density function rather than by a normal distribution; 2) that the impossibility to measure events shorter than the image acquisition time needs to be taken into account; 3) that the interaction time and run length of the motors can be estimated independent of the filament length distribution; and 4) that the dimeric nature of the motors needs to be considered when correcting for photobleaching. Moreover, our analysis reveals that controlling the temperature during the experiments with a precision below 1 K is of importance. We believe our method will not only improve the evaluation of experimental data, but also allow for better statistical comparisons between different populations of motor proteins (e.g., with distinct mutations or linked to different cargos) and filaments (e.g., in distinct nucleotide states or with different posttranslational modifications). Therefore, we include a detailed workflow for image processing and analysis (including MATLAB code), serving as a tutorial for the estimation of motility parameters in stepping-motility assays.

INTRODUCTION

Cytoskeletal motor proteins are essential for long-range intracellular transport (1), the malfunction of which can cause a number of pathologies including neurodegenerative diseases (2). The precise characterization of motor proteins with regard to their intrinsic function and the investigation of factors that influence their behavior thus constitutes an important part of medical and biophysical research. To study motor proteins in minimal in vitro systems, fluorescence imaging of single motors (stepping along their filamentous tracks) remains at the forefront of biophysical tools (3–6). However, quantitative estimation of the crucial motility parameters, namely velocity, interaction time, and run length, proves to be challenging owing to fundamental limitations in the experimental design. Because the processive run of a motor may be prematurely terminated by the end of a filament or the motor may be rendered invisible due to photobleaching of the attached fluorescent marker, so-called censored events will be part of any experimental data (7). As these censored events are prone to significantly bias the results, reliable correction methods are needed. Although corrections for finite filament length and photobleaching have been investigated individually in the past (8–12), the field still lacks a unified methodology. Here, we suggest an approach that addresses the above-mentioned challenges. Besides evaluating experimental data obtained from the motility of single, GFP-labeled kinesin-1 motors, we perform numerical simulations using a priori known parameters to show how statistical analysis allows for characterizing the certainty of a given measurement. Knowledge of the latter is of crucial importance when data from different measurements are to be compared. Along with our findings, we emphasize the importance of a precise temperature control during the measurements and describe

Submitted June 12, 2017, and accepted for publication September 21, 2017.
*Correspondence: stefan.diez@tu-dresden.de
Editor: E. Michael Ostap.
https://doi.org/10.1016/j.bpj.2017.09.024

a detailed workflow that facilitates the analysis of experimental data.

MATERIALS AND METHODS

Motor proteins and filaments

Histidine-tagged, truncated (1–430 amino acids) rat kinesin-1 labeled with eGFP (rKin430-eGFP) was expressed and purified as previously described (13). Porcine tubulin was purified from porcine brain (Vorwerk Podemus, Dresden, Germany) using established protocols (14). Microtubules were grown for 2 h at 37°C from a 80 μL BRB80 (80 mM PIPES; Sigma-Aldrich, St. Louis, MO), pH 6.9 adjusted with KOH (Merck, Kenilworth, NJ), 1 mM EGTA (Sigma-Aldrich), and 1 mM $MgCl_2$ (Merck) solution supplemented by 2 μM tubulin (75% unlabeled and 25% rhodamine-labeled TAMRA; Thermo Fisher Scientific, Waltham, MA), 1 mM GMP-CPP (Jena Bioscience, Jena, Germany), and 1 mM $MgCl_2$. A quantity of 80 μL of the microtubule solution was centrifuged in a Beckman Airfuge (A95 rotor; Beckman Coulter, Brea, CA) at 10,0000 $\times$ g for 5 min. The pellet was resuspended in a volume of 200 μL BRB80T (BRB80 supplemented by 1 mM taxol; Sigma-Aldrich). The solution was kept at room temperature overnight and the 200 μL microtubule solution was centrifuged and resuspended (in 200 μL BRB80T) again before the experiment.

Single molecule stepping assay

The employed stepping assays using total internal reflection fluorescence microscopy have been extensively described by Korten et al. (15). Briefly, we performed the experiments in flow channels (16), self-built from two glass coverslips (22 $\times$ 22 mm^2 and 18 $\times$ 18 mm^2; Corning, Corning, NY) that were cleaned in piranha solution (H_2O_2/H_2SO_2, 3:5; both purchased from Sigma-Aldrich), silanized with 0.05% dichlorodimethylsilane in trichloroethylene (Sigma-Aldrich) and glued together by heated pieces of Parafilm M (Pechiney Plastic Packaging, Chicago, IL). The flow sequence was as follows: 1) the flow channel was filled with a solution of TetraSpeck microspheres (diameter 100 nm; Thermo Fisher Scientific) diluted 200-fold in BRB80 that were used for drift and color correction (17). 2) After 2 min, the solution was exchanged with a BRB80 solution containing 77.5 μg/mL anti-β-tubulin antibodies (SAP4G5; Sigma-Aldrich). 3) After 5 min, the surface was blocked with a solution with 1% Pluronic F-127 (Sigma-Aldrich) in BRB80 for 15 min. 4) Microtubules diluted 10-fold to prevent microtubule intersections were incubated for 5 min to bind to the tubulin antibodies. 5) The microtubule solution was finally replaced by the motility solution—BRB80 containing 10 μM taxol, 0.04 mM glucose (Sigma-Aldrich), 0.2 mg/mL glucose oxidase (SERVA Electrophoresis, Heidelberg, Germany), 0.02 mg/mL catalase (Sigma-Aldrich), 10 mM DTT (Fermentas/Thermo Fisher Scientific), 0.1 mg/mL casein (Sigma-Aldrich), and 10 mM Mg-ATP (Sigma-Aldrich)—supplemented by 4 μg/mL rKin430-eGFP. For bleaching time estimation, undiluted sparsely labeled microtubules were used (10$\times$ less rhodamine labeling) and ATP was replaced with 10 mM AMP-PNP (Sigma-Aldrich) in all solutions.

Optical imaging

Fluorescence imaging was performed using an inverted fluorescence microscope (Observer Z1; Zeiss, Jena, Germany) with a 100$\times$ oil immersion objective (APOCHROMAT; numerical aperture 1.46; Zeiss) and an additional 1.33$\times$ magnifying Optovar (custom-build). The final pixel size was 117 nm. Microtubules were observed by epifluorescence using a Lumen 200 metal arc lamp (Prior Scientific Instruments, Fulbourn, United Kingdom) with a TRITC filter set (excitation 534/30, emission 593/40, dc R561; Chroma Technology, Rockingham, VT). rKin430-eGFP motor proteins were observed in total internal reflection fluorescence mode by using a PhoxX 488 nm Laser (Omicron-Laserage, Rodgau-Dudenhofen, Germany) with a GFP filter set (excitation 470/40, emission 525/50, dc 495; Zeiss) filter set. Image acquisition was performed at 100 ms exposure time in streaming mode by an electron-multiplied charge-coupled device camera (iXon Ultra DU-897U; Andor, Belfast, Northern Ireland) in conjunction with a MetaMorph imaging system (Universal Imaging, Downingtown, PA). The temperature was measured directly in the flow channel with a small temperature sensor (IT-23; Physitemp Instruments, Clifton, NJ) connected to a multipurpose thermometer (BAT-10; Physitemp Instruments). Temperature control was implemented using a custom-made hollow brass ring (MPI-CBG Mechanical Workshop, Dresden, Germany) around the objective connected to a water bath with combined cooling and heating unit (F25-MC Refrigerated/Heating Circulator; JULABO, Seelbach, Germany). Although the electric components of the microscope setup usually increase the room temperature as well as the temperature of the microscope body, our temperature control kept the temperature in the flow channel stable within 0.5 K over hours. See the Supporting Material for further experimental considerations.

Single molecule analysis

Single kinesin-1 molecules and microtubules were tracked using FIESTA (18). Molecules that showed any pauses or stalling were disregarded. After drift and color offset correction, the molecule position was projected on the microtubule centerline (see the Supporting Material for detailed instructions on tracking with FIESTA). The resulting distance along the centerline as well as the detachment position was utilized for further evaluation in different MATLAB scripts (The MathWorks, Natick, MA). For velocity simulations, a Monte Carlo simulation of a Poisson stepper was used, and to create exponential distributions, the MATLAB function "exprnd" was employed. Renormalization of the probability density function (PDF) was done with the Symbolic Math Toolbox of MATLAB and integration to obtain the cumulative distribution function (CDF) with the functions "int" and "matlabFunction". The evaluation of the cumulative distribution function utilized "ecdf", which also includes the optional Kaplan–Meier estimator; for least-square fitting, "fit" was used; and for maximum likelihood estimation, "mle" was employed. Bootstrapping was done with "parfor", included in the Parallel Computing Toolbox of MATLAB. The MATLAB code for the simulations as well as for the evaluation can be found within a compressed file in the Supporting Material (see description there).

Bootstrapping method

The statistical error of an evaluation method with a limited number of measurements can be estimated using a bootstrapping method (4). Briefly, from the data set, individual measurements are randomly selected with replacement. Here, the complete data set is always available when picking the measurement, which means that any measurement can also be selected more than once. Now, the new randomly selected data set is analyzed using the desired evaluation method. The procedure is repeated for sufficient number of repetitions (e.g., $n = 100$) with randomly selected data sets. The resulting bootstrapping distribution can be described by a normal distribution with the mean denoting the actual result and its SD describing the statistical error. This statistical error is only the result from random sampling and describes the error that is to be expected when repeating the experiment.

RESULTS

Single motor protein stepping assay

Motor proteins moving along their filaments can be described theoretically as Poisson steppers (Fig. 1). A simplified model of the motor protein kinesin-1 stepping

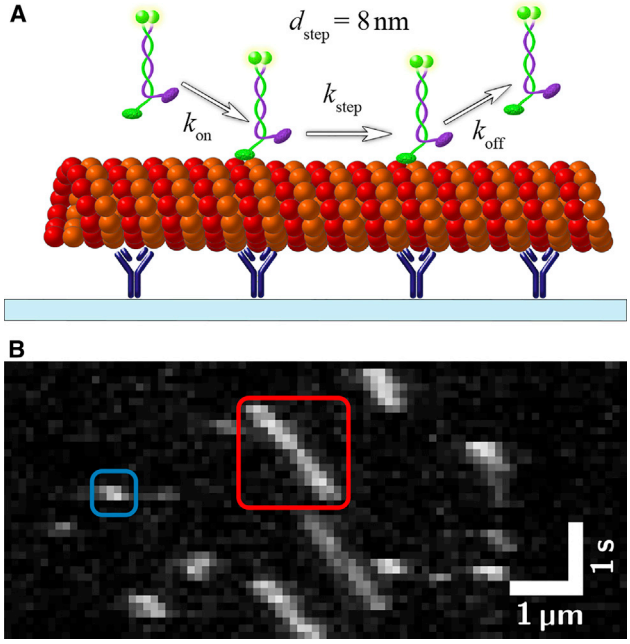

FIGURE 1 Single-motor stepping assay. (*A*) Shown here is a schematic depiction of a stepping assay. GFP-labeled kinesin-1 motors move along rhodamine-labeled microtubules that are immobilized by antitubulin antibodies to a glass surface. (*B*) Shown here is a typical kymograph (experimental data) of a stepping assay with kinesin-1. The red box shows a single motor that is measurable (interaction time $\tau = 1.3$ s); the blue box shows a motor with an interaction time that is too short for a reliable measurement. In the latter case, it is unclear whether the motor moved processively along the microtubule or interacted only unspecifically. To see this figure in color, go online.

along microtubules (MTs) is shown in Fig. 1 *A*. Whereas the attachment rate $k_{on} = k_{on}^0 \cdot$[Kinesin] is influenced by the motor protein concentration [Kinesin] in solution, the detachment rate k_{off} only depends on the motor-filament interaction. This interaction is described by 1) the motor velocity v, which can be used to derive the stepping-rate $k_{step} = v/d$ with d denoting the step size; 2) the interaction time τ, which can be used to derive the detachment rate $k_{off} = 1/\tau$; and 3) the run length R, which can be used to link the detachment rate k_{off} to the mechano-chemical cycle of the motor protein (e.g., $R = v \cdot \tau$). The first challenge in the determination of these parameters can be seen in the experimental kymograph in Fig. 1 *B*. There, clear linear motion can be observed for some motors (e.g., *red box*), but it is unclear if short interactions are actual movement or unspecific interactions (e.g., *blue box*). In our analysis, we therefore require a motor to be visible for five or more consecutive imaging frames and to move over a distance longer than the size of two pixels without pausing. Although these experimenter-defined thresholds appear arbitrary, we will show that their choice does not affect the results.

In the following, we will describe the procedures to estimate velocity, interaction time, and run length. Estimating the mean velocity will be rather straightforward, but we will show that it is more challenging (but equally important) to estimate the associated statistical error. As motor velocity is a good reporter of the environmental conditions (foremost temperature but also ionic strength and pH), knowledge about both mean velocity and error is of importance when potentially combining data from different fields of view or different experiments. Without taking precautions, the environmental parameters can vary significantly over the course of an experiment and unwanted influences can be minimized by pooling data from measurements with nondiffering velocities only. For reducing the systematic error in estimating the interaction time and run length, data analysis needs to address additional experimental challenges such as limited filament length and photobleaching. In addition to addressing the challenges that arise in single motor protein stepping assays, we provide extensive Supporting Material that includes the workflow for image processing in FIESTA, the MATLAB scripts used for the analysis, and many other experimental considerations (e.g., temperature control). Along with the proposed workflow for estimation of the motility parameters (see Workflow for Evaluating Velocity and Workflow for Evaluating Interaction Time and Run Length), this work can be used as a tutorial for investigating processive motor proteins. To facilitate the analysis, we incorporated the estimation of the motility parameters (including the proposed corrections) into FIESTA (version 1.6 or later).

Evaluation of the velocity

The velocity of individual processive motors is determined by tracking their positions over time, $X(t)$ and $Y(t)$. These positions are projected on the centerline of the filament and the distance $D(t)$ the motor moved along the filament is calculated. Fitting $D(t)$ with a linear function $D(t) = v \cdot t + c$ yields the velocity v. A typical experimental velocity distribution of $N = 543$ kinesin-1 motors is shown in Fig. 2 *A*. Although upon first sight the distribution resembles a normal distribution, hypothesis testing reveals that the data cannot be described by a normal distribution ($p < 0.001$, Kolmogorov-Smirnov test).

To investigate the reason behind the deviation from the normal distribution, we created a Monte Carlo simulation with 10,000 traces of motor proteins stochastically stepping with a rate of $k_{step} = 100\ s^{-1}$ and step size $d_{step} = 8$ nm ($k_{off} = 0$, total time per trace 20 s). We looked at the number of steps N_{steps} taken by each of these simulated motor proteins at specific time points (e.g., $t_1 = 1$ s, $t_2 = 3$ s, etc.). At each time point, N_{steps} is described by a Poisson distribution that can be approximated with normal distributions ($N_{steps} > 10$; Fig. S1 *A*). Because the velocity is described by $v = N_{steps} \cdot d_{step}/T$, the mean velocities are the same at each time point, but the widths of the normal distributions vary (Fig. S1 *B*).

If the detachment rate is changed to $k_{off} = 0.5\ s^{-1}$, each motor has a different interaction time, and thereby the

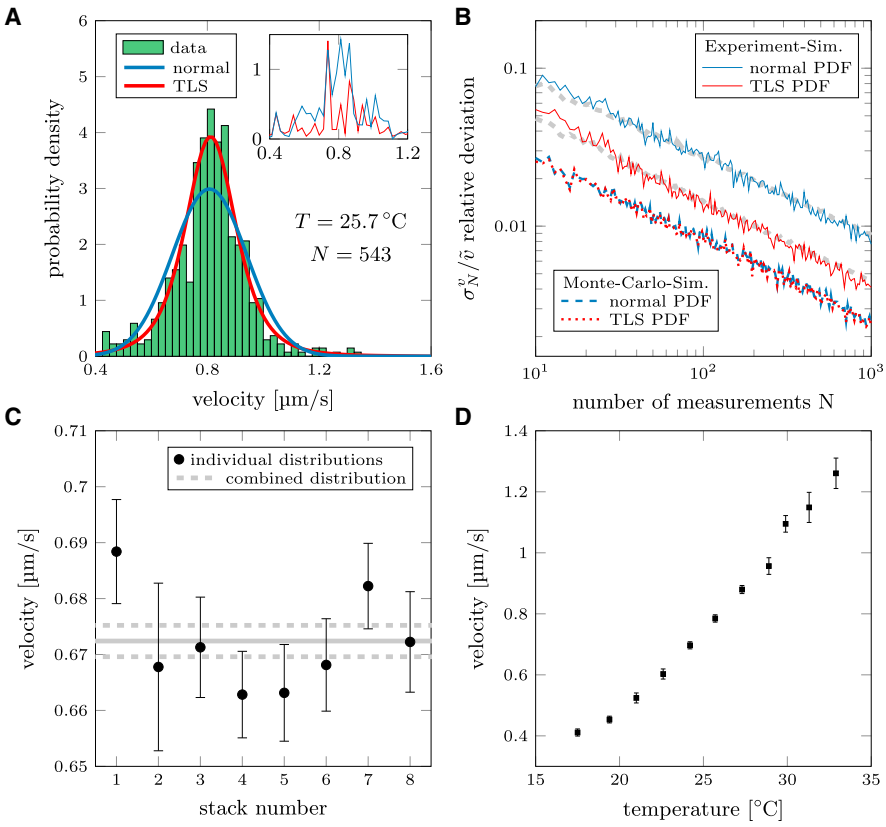

FIGURE 2 Evaluation of velocity in single-motor stepping assays. (*A*) Shown here is a histogram of kinesin-1's velocity distribution at temperature $T = 25.7°C$ with corresponding normal and TLS PDF (estimated with MLE). The inset shows the residuals between the estimated PDF and the experimental data. (*B*) Shown here is an evaluation of simulations using normal and TLS PDF. The red/blue dashed lines show the results using the Monte Carlo simulation whereas the solid lines depict results from simulations that resemble more realistic experimental conditions (Monte Carlo simulation with spatial averaging and additional positional error). Whereas both distributions yield the same precision for the results obtained from the Monte Carlo simulation, the TLS PDF is more precise than the normal PDF with simulated experimental data. The gray dashed lines are the average bootstrapping errors (see Materials and Methods), used to estimate the statistical errors. (*C*) Shown here is velocity during acquisition of one data set in eight different fields of view. The temperature was kept constant within 0.5 K (23.5–24°C) and the velocity shows only marginal deviations over time. Therefore, data can be pooled to create a combined data set (*gray lines*; *dashed lines* indicate error, $N_{\text{total}} = 5208$). (*D*) Shown here is dependence of velocity on temperature (measured in the flow channel). A temperature increase of 1 K increases the velocity by >5%. To see this figure in color, go online.

velocity distribution of all motors is a mix of normal distributions with the same mean values but different widths. In general, motors with shorter interaction times have a higher variance in the velocity distribution than motors that interact longer (under the same imaging conditions, Fig. S1 *B*). Consequently, the observed velocity distribution is not a normal distribution.

To further adjust the simulation to more realistic experimental conditions, spatial averaging over the positions $D(t)$ during the acquisition time of individual imaging frames (e.g., 100 ms accounting for a finite frame rate of $f = 10\ \text{s}^{-1}$) was performed and a positional error (due to tracking uncertainty) was incorporated by adding normally distributed noise ($\sigma = 100$ nm). Due to the finite acquisition time, no exact information about the attachment and detachment time can be extracted. Consequently, the positional averaging during acquisition of these frames would bias the estimation toward slower velocities (see the Supporting Material). Therefore, the first and last tracked frames have to be excluded in the linear regression of $D(t)$.

After obtaining a single velocity for each motor, a mean velocity v for one experiment can be obtained by estimating the characteristic parameters of a PDF with maximum likelihood estimation (MLE). As PDF, we used a *t* location-scale (TLS) distribution (19) that includes a shape parameter ν in addition to the location μ and scale σ parameters, which are conventionally used to describe a normal distribution. The TLS PDF that MATLAB uses in MLE is described by the following equation:

$$f(x\,|\,\mu,\sigma,\nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sigma\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left[\nu + \frac{\left(\frac{x-\mu}{\sigma}\right)^2}{\nu}\right]^{-\left(\frac{\nu+1}{2}\right)}. \quad (1)$$

As can been seen in Fig. 2 *A*, the TLS PDF fits the velocity distribution better than the normal distribution because the shape parameter ν accounts for heavier tails (see also Fig. S2 *A*). To compare the statistical errors when using these PDFs, we picked out a random set of *N* motor proteins from our simulation (with replacement, see bootstrapping in Materials and Methods). We calculated the velocity for each motor, and estimated the mean velocity v_N^i for that random set using both normal and TLS PDFs. The procedure was repeated $n = 100$ times and the deviation from the true velocity $\tilde{v} = 0.8$ µm/s ($k_{\text{step}} = 100\ \text{s}^{-1}$ and $d_{\text{step}} = 8$ nm) was calculated:

$$\sigma_N^v = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\overline{v}_N^i - \tilde{v}\right)^2};\ n = 100. \quad (2)$$

The relative deviation $\sigma_N^v/\tilde{v}$ is plotted as function of the number of measurements in Fig. 2 *B*. Although there is only a small difference between using a normal or TLS PDF when evaluating the Monte Carlo simulation, this difference increases when evaluating time-averaged traces with additional positional error (corresponding to experimental data). It turns out that, to reach a relative error $\Delta v/v$ of 2% (with $\Delta v = 2\sigma_N^v$, confidence interval 95%), only 200 measurements are necessary when using a TLS PDF compared to 750 measurements when using a normal PDF. Here, the difference between TLS PDF and normal PDF depends on the average number of data points in each trace available (defined by time resolution and interaction time) for each trace as well as on the positional error. On average, using either distribution yields the same mean velocity, but the TLS PDF yields a smaller relative error because it fits the velocity distribution significantly better (19) (see the Supporting Material for a more extensive investigation of the velocity evaluation).

In addition to measuring the velocity precisely, it is also essential to limit any external influence on the motility parameters, most importantly the temperature. Conventionally, velocity measurements on motor proteins are performed at "room temperature" and the actual temperature is not specified precisely. However, temperature variations can lead to small but significant velocity differences within one experiment and only by actively stabilizing the temperature in the flow channel. Using a temperature control system (within ± 0.5 K; see the Supporting Material), we could remove the temperature influence on the motor stepping (Fig. 2 *C*). In contrast, not accounting for temperature differences when comparing data sets (e.g., from different days or labs) can lead to a grave misinterpretation of data. Even a 1 K temperature increase can influence the velocity by >5% (Fig. 2 *D*). The temperature effect becomes even more prominent for measurements at room temperature (e.g., data points at 24.2–25.7°C) with ~13% increase in velocity over 1.5 K. Thus, measuring and reporting both temperature and velocity is of utmost importance when comparing the motility parameters of motor proteins. In turn, the motor velocity can be used as a control parameter to assure a constant temperature. In fact, together with our temperature control system, this strategy enabled us to pool experimental data sets from different image sequences and fields of view into a combined data set for further analysis.

Workflow for evaluating velocity

1) Extract the distance versus time information for each motor protein (see the Supporting Materials and Methods for instructions on how to track and analyze single fluorescent motor proteins using FIESTA (17)).
2) Fit distance versus time trace with linear regression leaving out first and last frame of the trace.
3) Use MLE to fit a TLS pdf to the velocity distribution.

To calculate the error of the mean velocity v with the TLS pdf, either use the 95% confidence interval (estimated with MLE) or bootstrapping (see Materials and Methods). The MATLAB code can be found within a compressed file (Data S1).

Evaluation of the interaction time and run length

Interaction time and run length of motor proteins are theoretically exponentially distributed (20) and our Monte Carlo simulations indeed show exponential distributions for both parameters. This can be attributed to a stochastic detachment with rate k_{off} where the interaction time is $\tau = {k_{\mathrm{off}}}^{-1}$ and the run length is $R = v \cdot {k_{\mathrm{off}}}^{-1}$. To test different methods to evaluate censored data, we simulated exponential distributions and evaluated them using three different methods previously described in the literature: 1) least-squares fitting of the probability density function (LSF-PDF) where the data is binned in a histogram, and the locations and heights of the bins are fitted by an exponential PDF $y_p = f(x\,|\,\mu) = \frac{1}{\mu}e^{-x/\mu}$ (20); 2) least-squares fitting of the cumulative distribution function (LSF-CDF) where the data is used to create a cumulative probability distribution that is fitted with the CDF $y_c = f(x\,|\,\mu) = 1 - e^{-x/\mu}$ (4); and 3) MLE, where the data is used directly to estimate the parameters of the exponential distribution. For complete exponential distributions (where all events are measurable), the MLE method yields the most precise results, because MLE can be solved analytically. In contrast, least-squares fitting involves numerical parameter optimization that is terminated when a certain tolerance is reached. The tolerance slightly decreases the precision with which the parameters are estimated. However, the experimental limitations prevent us from measuring complete exponential distributions. First, it is impossible to include motility events with τ shorter than the time resolution (in our case, 100 ms). Second, events with short interaction times might be easily discarded as noise during the evaluation procedure. Due to these missing short events, the measured exponential distributions are not complete and the evaluation method has to be adjusted: 1) using LSF-PDF, the first bin is underrepresented and needs to be disregarded when fitting the PDF (Fig. 3 *A*); 2) using LSF-CDF, a cutoff parameter x_0 has to be introduced in the CDF $y_c = f(x\,|\,\mu) = 1 - e^{-(x-x_0)/\mu}$ (4) to account for the missing measurements (Fig. 3 *B*), which cutoff parameter can be set as a constant (LSF-CDF(static) (4)) or as a free fit parameter (LSF-CDF(free) (5)); and 3) using MLE, a cutoff parameter can be employed to renormalize the PDF (21). For a truly single exponential PDF, the corrected parameter μ can then be estimated by subtracting the cutoff x_0 from all measurements and evaluating the data with MLE $(\mu = \sum(x_i - x_0)/n)$.

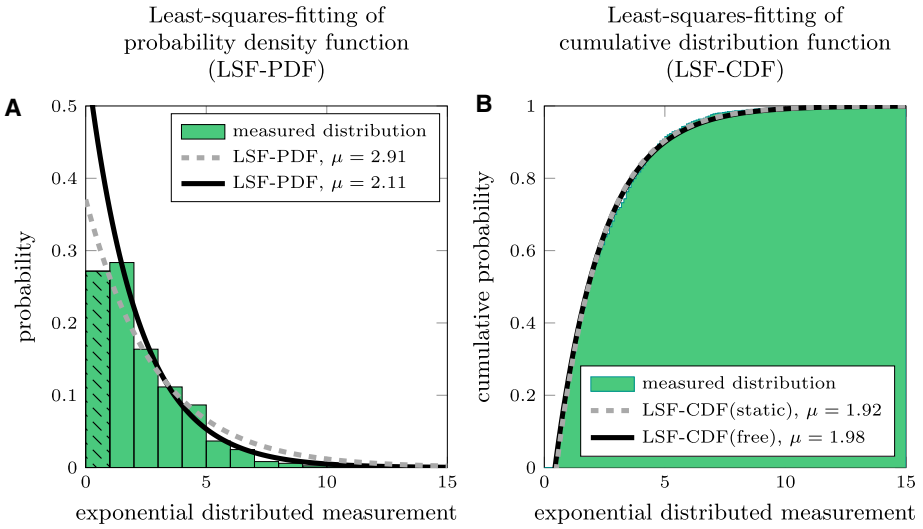

FIGURE 3 Modified exponential distribution to simulate real measurements. (*A*) Shown here is modified exponential probability distribution with $\mu = 2$ and $N = 1000$. Here, the first bar is underrepresented (*black pattern*), due to experimental limitations. The dashed line shows the LSF-PDF of the complete distribution, whereas the solid line shows the fit with the first bar excluded. (*B*) Shown here is modified cumulative probability of an exponential distribution with $\mu = 2$ and $N = 1000$. Here, all measurements $x < 0.5$ are disregarded, due to experimental limitations. The gray dashed line shows the LSF-CDF(static) of the distribution with a fixed x_0, whereas the solid black line shows the fit with x_0 as free fit parameter (LSF-CDF(free)). To see this figure in color, go online.

In the following, we characterize the evaluation of exponential distributions with LSF-PDF, LSF-CDF, and MLE by creating a random data set of an exponential distribution with $\tilde{\mu} = 2$ (we disregard the difference between run length and interaction time for now because the following applies to any exponentially distributed data). Analogous to the velocity in the previous section, we analyze the deviation of the estimated mean $\overline{\mu}$ (from $n = 100$ independent data sets) from the true $\tilde{\mu}$ in dependence upon the number of measurements N:

$$\sigma_N^{\mu} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\mu_N^i - \tilde{\mu}\right)^2}; \quad n = 100. \tag{3}$$

Fig. 4 *A* shows the results when using the complete exponential distribution. The relative deviation $\sigma_N^{\mu}/\tilde{\mu}$ depends only on the number of measurements due to the statistical nature of the exponential function. As expected, all methods show a clear $1/\sqrt{N}$ behavior, but the LSF-PDF yields a higher statistical error due to "coarse graining" by binning. Here, the data is binned but the distribution of the measurements within the bins is skewed (because an exponential distribution is continuously decreasing). Therefore, the data points for fitting the PDF differ slightly from the expected exponential distribution, which leads to the higher statistical error.

If the simulation is adjusted to represent realistic experimental results (Monte Carlo simulation with spatial averaging, additional positional error, and missing short events), two methods do not show the expected $1/\sqrt{N}$ behavior (see Fig. 4 *B*). Both LSF-CDF(static) and MLE can yield a systematic error because the cutoff is fixed and chosen rather arbitrarily. In experiments, however, the exact cutoff could be hidden within the limited time resolution or the limited tracking accuracy. For the example given in Fig. 4 *B*, the shortest interaction time was 0.5 s (at least five frames with a time resolution of 100 ms), but because the interaction can only be measured for full frames, a number of motors with interaction times between ~0.45 and

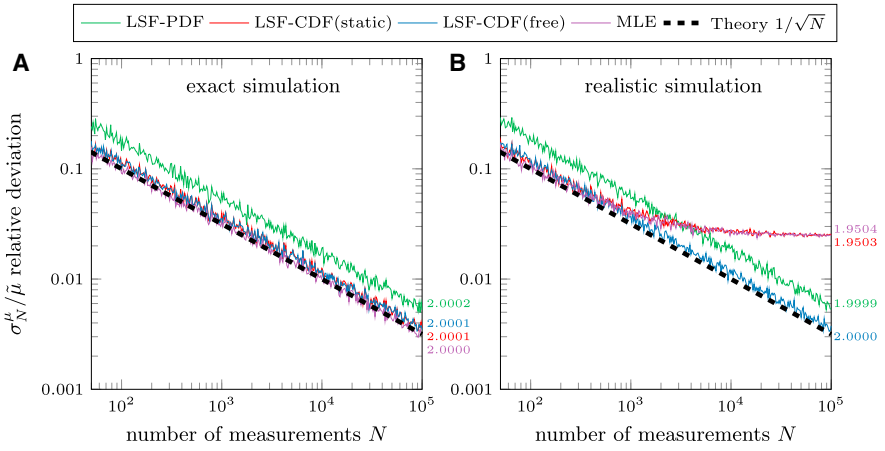

FIGURE 4 Comparison of methods to estimate the parameters of exponential distributions. (*A*) Shown here is relative deviation $\sigma_N^{\mu}/\tilde{\mu}$ from the a priori value of the simulation estimated using an exact and complete exponential distribution. All methods reach the right result (see values on the *right*); only the LSF-PDF method has a higher statistical error due to coarse-graining by binning. (*B*) Shown here is the same relative deviation $\sigma_N^{\mu}/\tilde{\mu}$, but using a modified exponential distribution with a resolution of 0.1 μm or s, only $x \geq 0.5$. LSF-CDF(static) and MLE methods with cutoff $x_0 = 0.5$ fail because the real cutoff should be at $x_0 = 0.45$ (see the Supporting Material). All simulations used $\tilde{\mu} = 2$ and $n = 100$ (for LSF-PDF, a 0.5 bin width was used). Values on the right are mean results of simulations using $\tilde{\mu} = 2$, $N = 10^5$, and $n = 10^4$. To see this figure in color, go online.

0.5 s will also be also present in the distribution. Therefore, the correct cutoff should actually be smaller than 0.5 s (more discussion on choosing the cutoff in the Supporting Material). The LSF-PDF method is not influenced by the missing short events as long as all bins that are affected by the missing events are disregarded, but again yields a higher statistical error due to coarse-graining. LSF-CDF(free) avoids choosing a fixed cutoff x_0 (by adding it as an additional fit parameter) and yields precise results for the modified distribution. Admittedly, employing an additional fit parameter comes at the price that it may distort the results when evaluating more complex distributions (e.g., double exponential). Note that the statistical error cannot be smaller than $\mu/\sqrt{N}$, even if the confidence interval of the fitting is smaller. This fitting error from least-squares-fitting occurs in addition to the statistical error that results from the random sampling of the exponential distribution. Hence, it is more important to measure a large number of data points rather than improving the precision of the individual measurements. The code for fitting a particular experimental data set can be found within a compressed file in the Supporting Material, along with an extensive discussion on the creation of the CDF, the least-squares-fitting, and the cutoff issue. In the following, we only use LSF-CDF(free) to evaluate exponential distributions. We note that MLE would work similarly (including correction for censored events). For comparison, we included both evaluation methods in the MATLAB code and also compared their statistical errors using simulations (see the Supporting Material).

Correction for finite filament length

Because each filament has a finite length, some motor proteins are destined to run into the end of their track and detach (Fig. 5 *A*). These events influence the measurement of run length (or interaction time) and introduce a dependence on the filament length (8). Therefore, identical motor proteins moving along longer filaments would have a higher observed run length (or interaction time) than motors stepping on short filaments. Here, we present a correction for these so-called "end-events" by using the Kaplan–Meier estimator (22) in LSF-CDF(free) to adjust the cumulative probability distribution for these censored events. Because it is possible to image the filaments and track single motor proteins, the detachment positions along the filament can be determined. Any event where a motor protein detaches near the end of a filament (within one pixel), is then scored as end-event (the exact procedure to calculate the adjusted cumulative probability distribution can be found in the Supporting Material). To verify the proposed method, we simulated events of motor proteins landing on filaments with a random length distribution (assuming a Schulz distribution (23), see equation in the Supporting Material) and assessed which motors reach the filament end. These traces are included in the analysis as censored events.

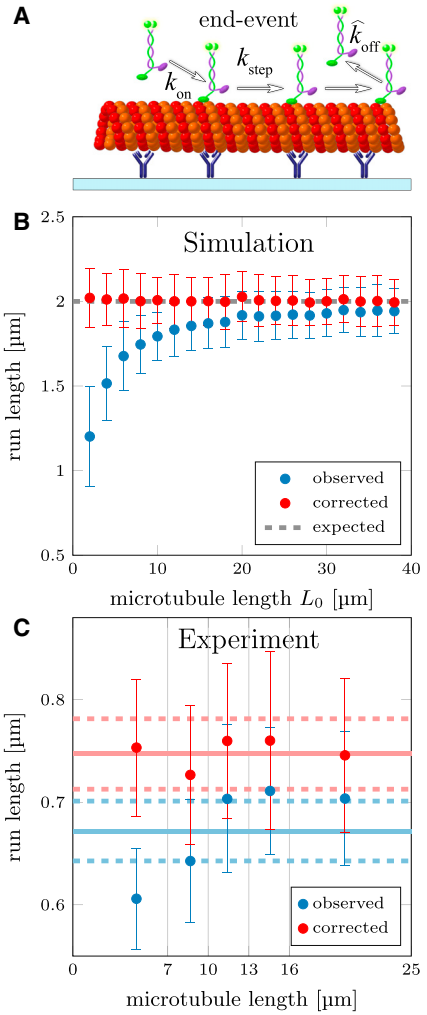

FIGURE 5 Correction for finite filament length. (*A*) Shown here is a schematic depiction of GFP-labeled kinesin-1 motors that reach the end of the microtubule and are forced to detach there (scored as "end-event"). (*B*) Shown here is the simulation of motor proteins landing on 10 filaments, which are picked out of a Schulz distribution with L_0. Some motors reach the end and detach prematurely, resulting in underestimation of the run length. Using the Kaplan–Meier estimator (for end-events), a corrected cumulative probability distribution can be calculated and LSF-CDF(free) is used to estimate a corrected run length (the same can be applied to the interaction time). (*C*) Shown here is experimental data of kinesin-1 run length, where traces were separated in five groups according to the length of their filament ($N \approx 1000$). Solid line shows mean run length of combined distribution with bootstrapping error (*dashed lines;* see Materials and Methods). To see this figure in color, go online.

Fig. 5 *B* compares the run lengths from our simulated data with and without correction. Note that neglecting the length correction leads to a systematic error that influences the measurement because a certain number of motor proteins will always reach the end even for long filaments. The method was also verified experimentally when we tracked 5208 kinesin-1 motor proteins stepping along MTs (same data as in Fig. 2 *C*) in eight different fields of view (temperature 23.5−24°C). Fig. 5 *C* shows the run lengths grouped

according to the length of their MTs (combined data set was separated in groups of $N \approx 1000$). A dependence of the observed run length on the MT length is seen in the uncorrected data, whereas the correction using the Kaplan–Meier estimator yields the same run length for all groups as well as for the combined distribution. The errors can be estimated using bootstrapping, where measurements are randomly selected (with replacement, see Materials and Methods), classified for end-events and then evaluated using LSF-CDF(free) with the Kaplan–Meier estimator. After sufficient repetitions ($n = 100$), the resulting bootstrapping distribution yields the mean run length $\overline{R}$ and the statistical error after correction $\Delta R = 2\sigma_R$. Therefore, any influence of the Kaplan–Meier estimator on the statistical error can be accounted for. An alternative method to verify length correction can be found in the Supporting Material.

Correction for photobleaching

Another experimental limitation is the statistical nature of photobleaching after a fluorophore has emitted a certain number of photons. Photobleaching influences the measurement of the observed interaction time (or run length) and introduces a dependence on the bleaching rate k_{bleach} (see Fig. 6 *A*). Even though the lifetime of the fluorophores can be increased by adding antifade solutions (15), the effect of photobleaching cannot be eliminated fully in the experiments. Here, we describe the bleaching probability as a combination of one- and two-fluorophore bleaching (see Eq. 4), because even though dimeric, GFP-labeled motor proteins are always tagged with two fluorophores, not all of these fluorophores are active (see the Supporting Material):

$$P_{\text{bleach}} = (2-\rho)k_b e^{-k_b x} + 2(\rho-1)k_b e^{-2k_b x} \ldots k_b = k_{\text{bleach}}, \tag{4}$$

$$P_{\text{stepping}} = (2-\rho)\left(k_{\text{off}}+k_b\right)e^{-\left(k_{\text{off}}+k_b\right)x} + (\rho-1)\left(k_{\text{off}}+2k_b\right)e^{-\left(k_{\text{off}}+2k_b\right)x}. \tag{5}$$

In Eq. 4, the parameter ρ denotes the fraction of motors with only one active fluorophore. Because combining the evaluation of detachment and photobleaching is not trivial (Eq. 5) and the corresponding addition of more parameters into the LSF-CDF leads to unstable solutions, we introduce a different approach by assigning a bleaching probability to each individual motor protein according to its interaction time. Afterwards, the data is analyzed with a single exponential using LSF-CDF(free) several times ($n = 100$) and in each iteration different events will be randomly scored as bleaching events in agreement with their bleaching probability. Combined with the end-events, these censored events are corrected for by using the Kaplan–Meier estimator. This way, the bleaching correction is averaged over many iterations of the run length or interaction time estimation. Additionally, the data can be resampled in every iteration to combine photobleaching correction with the bootstrapping method that now not only yields a corrected measurement, but also the statistical error of the result.

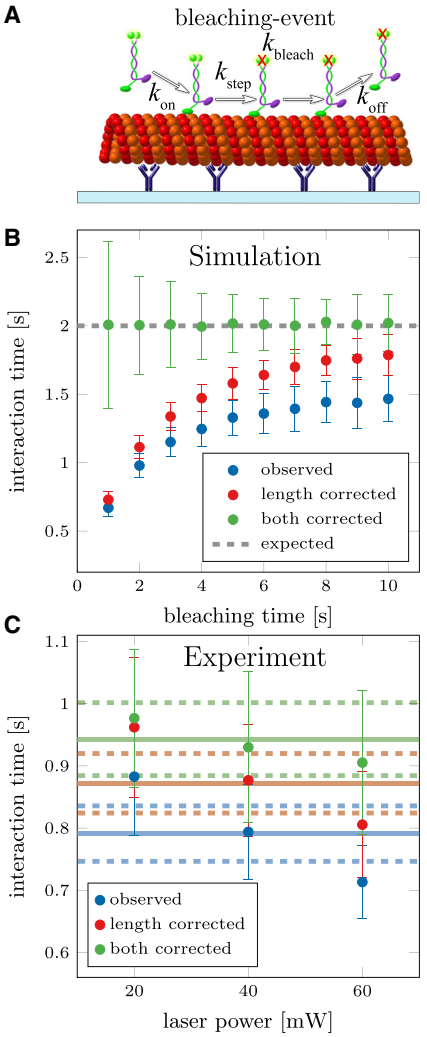

FIGURE 6 Correction for photobleaching. (*A*) Shown here is a schematic depiction of GFP-labeled kinesin-1 motors that photobleach while moving along the microtubule. The observed detachment rate is influenced by photobleaching. (*B*) Shown here are simulations with finite filament lengths (Schulz distribution $L_0 = 5$ μm) and photobleaching of motor proteins with one or two fluorophores ($\tau_{\text{bleach}} = 1 - 10$ s, $\rho = 0.5$). Distribution of bleaching times (ρ and k_{bleach}) is measured from a different population of simulated motor proteins (e.g., immobilized motors). Simulations used $\tilde{\tau} = 2$ s, $N = 1000$, and $n = 100$. (*C*) Shown here are experimental data from kinesin-1 at different laser excitation intensities (laser power at controller). The bleaching time distribution was measured with immobilized motor proteins (in the presence of AMP-PNP) in a second flow channel on the same coverslip (same settings and solutions). Here, bleaching rates k_{bleach} and ratio ρ were measured for each excitation power: $k_{\text{bleach}} = 0.041 \pm 0.003$ s^{-1} and $\rho = 0.41 \pm 0.09$ for 20 mW; $k_{\text{bleach}} = 0.13 \pm 0.02$ s^{-1} and $\rho = 0.32 \pm 0.12$ for 40 mW; and $k_{\text{bleach}} = 0.22 \pm 0.02$ s^{-1} and $\rho = 0.43 \pm 0.11$ for 60 mW. To see this figure in color, go online.

To verify the proposed correction, we extended our simulations to include photobleaching with a mixture of one- and two-fluorophore bleaching, $\rho = 0.5$. Fig. 6 *B* shows the dependence of the observed and corrected interaction times on the bleaching time. Here, only when correcting for both finite filament length and photobleaching, the expected interaction of $\tilde{\tau} = 2$ s was estimated. We tested the photobleaching correction experimentally by imaging single kinein-1 motor proteins in stepping assays at different laser excitation intensities. Here, for each intensity, the bleaching rate k_{bleach} as well as ratio ρ was measured in a second flow channel (on the same coverslip) by immobilizing GFP-labeled kinesin-1 on MTs with AMP-PNP (a nonhydrolysable analog of ATP). The evaluation of the photobleaching using FIESTA is described in the Supporting Material. Results in Fig. 6 *C* validate the correction for photobleaching, because the dependence of the interaction time on the laser intensity is removed and the estimation of a combined interaction time is possible.

Workflow for evaluating interaction time and run length

1) Extract the interaction time and run length information for each motor protein (see the Supporting Materials and Methods for instructions on how to track and analyze single fluorescent motor proteins using FIESTA).
2) Score end-events according to their detachment position in relation to the filament end.
3) Estimate bleaching probability from the separate channel with immobilized motors (see the Supporting Materials and Methods for photobleaching measurement with FIESTA).
4) Assign bleaching-events, create cumulative probability distribution (censored end- and bleaching-events) with the Kaplan-Meier-Estimator and use LSF-CDF (free) to estimate interaction time and length.
5) Bootstrapping: repeat step 4 with different randomly selected traces. Reassign bleaching events randomly in each iteration according to the bleaching probability.

The MATLAB code for evaluation of velocity, bleaching time, interaction time, and run length can be found within a compressed file (Data S1; description in the Supporting Materials and Methods). Note that censored events also include events where the motor proteins move in or out of the field of view, as well as traces that start or end in the first or last imaging frame, respectively.

DISCUSSION

We presented a method to investigate the motility parameters of single, fluorescently labeled motor proteins in stepping assays, including experimental enhancements, software for tracking and analysis, and precise evaluation of the measurements. For verification, simulations were performed to check whether the methods yield the true results and different methods for evaluation and correction were compared with respect to their systematic and statistical error. The best methods were then used to characterize the temperature dependence of the velocity as well as the influence of the filament length and photobleaching on interaction time and run length experimentally. Furthermore, correction methods are proposed to minimize the influence of the experimental setup on the obtained results. These corrections are shown to work in simulations as well as with experimental data and show the advantage to previously used methods for the evaluation of data from single fluorescent motor proteins.

First, we found that a measured velocity distribution from single motor proteins cannot be described as a normal distribution. When assuming a simple Poisson–Stepper model the velocity of motors with longer interaction times can be estimated more precisely then for short events, which leads to deviations from the normal distribution due to heavier tails. Still, the normal distribution is commonly used when evaluating velocity distributions (3–6), even though Norris et al. (6) already show clear deviations (Fig. S2). Here, we propose using a TLS distribution for estimation of the mean velocity because the TLS PDF fits better to the simulated as well as the experimentally measured velocity distributions. Compared to the normal distribution the TLS method yields a smaller statistical error. Because it is possible to precisely measure the velocity ($\Delta v/v < 0.01$ with $N > 1000$) this motility parameter can be used as a control parameter to verify that the temperature indeed was stable even without the information of the additional temperature sensor in the flow channel.

Second, we compared different methods to evaluate exponential distributions and we found that using the LSF-CDF yields the best results. Here, the introduction of a cutoff x_0 as a free fit parameter is sufficient to account for missing short events. We note that the MLE method can also be used if the PDF is renormalized to account for the missing short events (21), but the a priori knowledge of the cutoff parameter can essentially introduce a systematic error and bias the results. Nonetheless, because MLE can also account for censored events, the LSF-CDF(free) method can easily be replaced by MLE. (Note: The MATLAB code in the Supporting Material includes both methods.) We compared both methods using simulations (see the Supporting Material) and found that MLE has a slightly smaller statistical error (mainly due to one less fitting parameter). However, when evaluating noisy data (simulated or experimental) qualitatively the fit of the PDF to the distribution was strongly influenced by the choice of the cutoff parameter. While the actual method to be used is a matter of choice we recommend using LSF-CDF(free) or extending MLE to allow for estimation of the best cutoff parameter (e.g., by machine learning).

Third, we propose specific corrections for finite filament length and photobleaching using the Kaplan–Meier estimator with the LSF-CDF(free) method. Because we can track both the fluorescently labeled motors and the

filaments, it is possible to determine end-events, which censor some specific events in a data set. By using the Kaplan–Meier estimator we can adjust the CDF for these end-events and thereby correct for the finite filament length. The advantage over previously published methods (8), which use a correction term that includes the average filament length of the experiment, is that the underlying length distribution of the filaments does not influence the evaluation. Therefore, it does not matter if motor proteins are stepping on one particular filament or on a random set of filaments. In addition, variations in filament length between different fields of view do not affect the results. Correction for photobleaching is not as trivial as the filament length correction. Previously, corrections for photobleaching assumed the simple relation $k_{\mathrm{observed}} = k_{\mathrm{off}} + k_{\mathrm{bleach}}$ (9,24), with k_{observed} describing the observed detachment rate being the superposition of the real detachment rate k_{off} and the bleaching rate k_{bleach}. There, by either estimating the bleaching rate (9) or extrapolating k_{observed} using different laser intensities (24), a corrected interaction time could be calculated. Unfortunately, these methods fail when the molecules of interest are labeled with more than one fluorophore as is the case with dimeric motor proteins where a GFP is expressed on each monomeric motor unit. Most importantly, usually not all fluorophores are active and a mixture of motor proteins with either one or two active fluorophores is observed (12). Therefore, we propose to measure the bleaching time as well as the ratio of one to two fluorophore bleaching to calculate the bleaching probability for a specific experiment. Over many iterations, we can now randomly assign bleaching-events according to the interaction time of the motor proteins. By including them in the censored events, our correction for both, photobleaching and finite filament length, is achieved. Consequently, our simulations show slight deviations from the expected true values only for bleaching times in the range of—or shorter than—the interaction time.

How well does our proposed method perform when 1) the interaction time of the motor is significantly longer than the bleaching time of the fluorophores, 2) the run lengths of the motors are significantly larger than the filament lengths, and 3) the motors exhibit frequent switching between movement and stalling? With regard to the limitations originating from photobleaching, adjustment of the imaging conditions (e.g., exposure time and light intensity, such that the bleaching rates are lower than the motor detachment rates) are effective means to reduce the systematic error. Furthermore, novel photostable fluorescent proteins as well as functionalized fluorophores (e.g., GFP-boosters based on nanobodies) will reduce the influence of photobleaching. In any case, comparing the motility parameters obtained under different imaging conditions can directly show if photobleaching influenced the results. With regard to investigating highly processive motor proteins, which reach the filament ends in most of the cases, the only solution is to use longer MTs. Tweaking the protocols for MT polymerization (e.g., by extending the growths times in conjunction with lowering the tubulin concentration) allows for the generation of MTs with lengths above 50 μm, sufficient to reliably determine the run lengths in any cellular context. Moreover, discarding all motility events from filaments with lengths below a threshold is possible, because our proposed correction algorithms do not require a certain MT length distribution. In general, the more motor traces are censored the less accurate the precision will be. With respect to super-processive motility (such as Kinesin-3 (25) and Kinesin-8 (26)) we note that the run length is not a suitable motility parameter and rather other measures (such as the motor's end-reach probability in dependence of the landing position) should be applied. With regard to motors that exhibit frequent stalling, our corrections for finite filament length and photobleaching do stay valid (Kaplan–Meier estimator can still be used), but the motors cannot be described by simple Poisson steppers anymore. Therefore, the underlying models will have to be adjusted accordingly, e.g., by using double-exponential functions for interaction time and run length.

We conclude, to precisely characterize the stepping of motor proteins on their filaments, the following steps are necessary:

1. The temperature should be stable throughout the experiments to combine and evaluate many traces at the same condition in one data set. Because even small changes in temperature influence the motility parameters significantly, it is essential to measure the assay temperature precisely. Here, measuring and specifying the room temperature is not sufficient as the actual temperature in the flow channel can be up to 3 K higher due to microscope-internal heat sources, such as electrical components and light sources. Hence, any results given for velocity, run length, and interaction time should include the temperature in the flow channel within $\pm$ 1 K. Furthermore, to investigate different motor or filament populations, we recommend incorporating them in the same flow channel or at least on the same coverslip to minimize any temperature differences in the experiments. A detailed description of the temperature control can be found in the Supporting Material, which is easy to implement using any mechanical workshop (SolidWork files are available on request).
2. Corrections for finite filament length and photobleaching need to be included, because otherwise both interaction time and run length are underestimated (e.g., see Zimmermann et al. (27)). Here, the systematic errors can be on the same order as the statistical error and therefore careful consideration of the evaluation method is essential. For that reason, we provide an extensive description of the evaluation method, including the MATLAB code, to efficiently measure a sufficient number of motor proteins (to reduce the statistical error)

and to address limitations in the design of the experimental assay (to remove the systematic error).

3. The motility parameters velocity, interaction time, and run length should always be estimated when comparing different data sets. Motor proteins could have the same run length, but different velocities (k_{step}) and interaction times (k_{off}), so only comparing one motility parameter might result in the misconception that even if the experiment yields the same result for different motors, the underlying motility mechanism could still be different.

In summary, the described evaluation should allow for a better statistical comparison of motor proteins influenced by external factors, e.g., ionic strength, ATP concentration, nucleotide state of the filaments, or posttranslational modifications of the filaments. Furthermore, comparison of different motor proteins as well as motor populations, e.g., structural differences or binding of regulatory proteins, will then become possible. We believe the methodology developed in our work will provide a reliable framework for the evaluation of a wide range of experiments with single fluorescently labeled motor proteins.

SUPPORTING MATERIAL

Supporting Materials and Methods, eight figures, three tables, and one data file are available at http://www.biophysj.org/biophysj/supplemental/S0006-3495(17)31036-6.

AUTHOR CONTRIBUTIONS

F.R. and S.D. conceived and designed the experiments. F.R. and L.K. performed the experiments and analyzed the data. F.R. contributed reagents/materials/analysis tools. F.R. and S.D. wrote the article.

ACKNOWLEDGMENTS

We thank Friedrich Schwarz for help with the design and the mechanical workshop at the Max Planck Institute for Molecular Cell Biology and Genetics (MPI-CBG) for manufacturing the cooling/heating ring for the objective; Corina Bräuer for technical support; Aniruddha Mitra, Jens Ehrig, Rahul Grover, and Georg Krainer for comments on the manuscript; and the whole Diez Lab for fruitful discussions.

We acknowledge financial support from the European Research Council (Starting Grant 242933), the Deutsche Forschungsgemeinschaft (DFG) through the Center for Advancing Electronics Dresden (CFAED), and the Technische Universität Dresden.

REFERENCES

1. Vale, R. D. 2003. The molecular motor toolbox for intracellular transport. *Cell.* 112:467–480.
2. Morfini, G., N. Schmidt, …, S. Kins. 2016. Conventional kinesin: biochemical heterogeneity and functional implications in health and disease. *Brain Res. Bull.* 126:347–353.
3. Vale, R. D., T. Funatsu, …, T. Yanagida. 1996. Direct observation of single kinesin molecules moving along microtubules. *Nature.* 380:451–453.
4. Thorn, K. S., J. A. Ubersax, and R. D. Vale. 2000. Engineering the processive run length of the kinesin motor. *J. Cell Biol.* 151:1093–1100.
5. Walter, W. J., V. Beránek, …, S. Diez. 2012. Tubulin acetylation alone does not affect kinesin-1 velocity and run length in vitro. *PLoS One.* 7:e42218.
6. Norris, S. R., M. F. Núñez, and K. J. Verhey. 2015. Influence of fluorescent tag on the motility properties of kinesin-1 in single-molecule assays. *Biophys. J.* 108:1133–1143.
7. Gijbels, I. 2010. Censored data. *WIREs Comp. Stat.* 2:178–188.
8. Thompson, A. R., G. J. Hoeprich, and C. L. Berger. 2013. Single-molecule motility: statistical analysis and the effects of track length on quantification of processive motion. *Biophys. J.* 104:2651–2661.
9. Pierce, D. W., and R. D. Vale. 1998. Assaying processive movement of kinesin by fluorescence microscopy. *Methods Enzymol.* 298:154–171.
10. Kawaguchi, K., and S. Ishiwata. 2000. Temperature dependence of force, velocity, and processivity of single kinesin molecules. *Biochem. Biophys. Res. Commun.* 272:895–899.
11. Böhm, K. J., R. Stracke, …, E. Unger. 2000. Effect of temperature on kinesin-driven microtubule gliding and kinesin ATPase activity. *FEBS Lett.* 466:59–62.
12. Lakämper, S., A. Kallipolitou, …, E. Meyhöfer. 2003. Single fungal kinesin motor molecules move processively along microtubules. *Biophys. J.* 84:1833–1843.
13. Rogers, K. R., S. Weiss, …, R. Cross. 2001. KIF1D is a fast non-processive kinesin that demonstrates novel K-loop-dependent mechanochemistry. *EMBO J.* 20:5101–5113.
14. Castoldi, M., and A. V. Popov. 2003. Purification of brain tubulin through two cycles of polymerization-depolymerization in a high-molarity buffer. *Protein Expr. Purif.* 32:83–88.
15. Korten, T., B. Nitzsche, …, S. Diez. 2011. Fluorescence imaging of single kinesin motors on immobilized microtubules. *Methods Mol. Biol.* 783:121–137.
16. Gell, C., V. Bormuth, …, J. Howard. 2010. Microtubule dynamics reconstituted in vitro and imaged by single-molecule fluorescence microscopy. *Methods Cell Biol.* 95:221–245.
17. Leduc, C., F. Ruhnow, …, S. Diez. 2007. Detection of fractional steps in cargo movement by the collective operation of kinesin-1 motors. *Proc. Natl. Acad. Sci. USA.* 104:10847–10852.
18. Ruhnow, F., D. Zwicker, and S. Diez. 2011. Tracking single particles and elongated filaments with nanometer precision. *Biophys. J.* 100:2820–2828.
19. Hughes, J., S. Shastry, …, J. Fricks. 2013. Estimating velocity for processive motor proteins with random detachment. *J. Agric. Biol. Environ. Stat.* 18:204–217.
20. Block, S. M., L. S. B. Goldstein, and B. J. Schnapp. 1990. Bead movement by single kinesin molecules studied with optical tweezers. *Nature.* 348:348–352.
21. Woody, M. S., J. H. Lewis, …, E. M. Ostap. 2016. MEMLET: an easy-to-use tool for data fitting and model comparison using maximum-likelihood estimation. *Biophys. J.* 111:273–282.
22. Kaplan, E. L., and P. Meier. 1958. Nonparametric estimation from incomplete observations. *J. Am. Stat. Assoc.* 53:457–481.
23. Jeune-Smith, Y., and H. Hess. 2010. Engineering the length distribution of microtubules polymerized in vitro. *Soft Matter.* 6:1778–1784.
24. Mashanov, G. I., D. Tacon, …, J. E. Molloy. 2003. Visualizing single molecules inside living cells using total internal reflection fluorescence microscopy. *Methods.* 29:142–152.
25. Soppina, V., S. R. Norris, …, K. J. Verhey. 2014. Dimerization of mammalian kinesin-3 motors results in superprocessive motion. *Proc. Natl. Acad. Sci. USA.* 111:5562–5567.
26. Varga, V., J. Helenius, …, J. Howard. 2006. Yeast kinesin-8 depolymerizes microtubules in a length-dependent manner. *Nat. Cell Biol.* 8:957–962.
27. Zimmermann, D., A. Santos, …, R. S. Rock. 2015. Actin age orchestrates myosin-5 and myosin-6 run lengths. *Curr. Biol.* 25:2057–2062.

Biophysical Journal, Volume *113*

Supplemental Information

Challenges in Estimating the Motility Parameters of Single Processive Motor Proteins

Felix Ruhnow, Linda Kloβ, and Stefan Diez

SUPPORTING MATERIAL
Challenges in estimating the motility parameters of single processive motor proteins

F. Ruhnow, L. Kloß, S. Diez, Felix Ruhnow[a], Linda Kloß[a], Stefan Diez[a,b,*]

[a] *B CUBE - Center for Molecular Bioengineering, Technische Universität Dresden, Dresden, Germany*
[b] *Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany*

Contents

*Corresponding author: stefan.diez@tu-dresden.de

S1. Experimental considerations

When measuring single fluorescently-labeled motor proteins in stepping assays the following experimental enhancements should be considered:

1. Tetraspeck Microspheres (Invitrogen) should be used as reference beads not only for drift correction but also to account for any color offset between the filament image(s) and the motor protein images. The 0.1 µm diameter beads are recommended for GFP labeled motors and the 0.2 µm diameter beads are suitable for brighter probes like quantum dots. The color offset also includes drift that occurs between the acquisition of the images, e.g. after switching filter sets.
2. The temperature should either be measured directly (e.g. by adding a probe to the sample) or by using the velocity as a reference (e.g. after control measurement at different temperatures). Note that the temperature in the flow channel might deviate from room temperature (depending on the setup). In our experiments the temperature was up to 3 K higher than the room temperature (mostly due heating of the microscope stand by electrical components as well as light sources). Additionally, the temperature most likely increases during the experiment, due to heat generated by light sources, computers and other electronic equipment as well as humans. This temperature increase can be easily mistaken as a significant difference e.g. when measuring one population first and the next population after some time, with a higher temperature in the flow channel.
3. Temperature control of the objective is sufficient. The oil between the TIRF objective and coverslips acts as a good heat conductor and the measured temperature in the flow channel near the objective almost matches the temperature of the water bath. A stable temperature of the objective also decreases defocusing effects due thermal expansion within the objective and therefore longer image acquisition without auto-focus or manual refocusing is possible.
4. The filaments should be imaged before the motor proteins, because long laser exposure of the filaments during imaging of the motor proteins will lead to photobleaching of the fluorescently-labeled filaments and thereby reduce the accuracy of the filament tracking. Normally, one image of the filaments is sufficient, but the tracking accuracy of the centerline can be increased by averaging over several frames (centerline position can be averaged).
5. Choose appropriate filament density to reduce crossings of filaments. While it is possible to track filaments crossings, the accuracy is reduced and the matching of motor proteins to their respective filaments becomes problematic. When adding the filaments in the flow channel, high flow speeds with less incubation time align the filaments parallel on the surface and reduce crossings while maintaining a high filament density.

S2. Tracking with FIESTA

The workflow to track single fluorescently-labeled motor proteins with respect to their filaments was optimized greatly in order to analyze sufficient events for statistical comparison. FIESTA (version 1.05.0005) now includes multi-channel evaluation to improve data analysis and user interaction. The following steps should be used in order to analyze many tracks efficiently:

1. **Open Stack Special** (Menu→Data), choose files for motor proteins (Channel 1 recommended) and filaments (other Channels).
2. Track filaments and reference beads in filament channel. *Check correct focus drift checkbox in the filament option.*
3. Track motor proteins and reference beads in molecule channel. Note, motor proteins and beads should be tracked at two different threshold levels. *Choose appropriate maximum velocity in the Configuration (will reduce time for feature connection but could lead to unconnected tracks if set too low).*
4. **Load Tracks** (Menu→Data) from reference beads in the molecule channel, use **Find Molecules for Drift Correction** (Menu→Edit) or manually choose reference beads for drift correction. Use **Set Drift** (Right-Click on Molecules Panel) to calculate drift and check **Subtract Drift** checkbox in the Molecules Panel (**Save Drift** (Menu→Options) file is recommended). **Save selected Tracks** (Menu→Edit) using only the selected drift corrected reference beads.
5. **Load Tracks** from single motor proteins and select molecule tracks co-localizing with filaments (evaluating motor tracks for each filament separately is recommended). Verifying motor protein tracks is best done using the kymograph tool in the Tools-Scan Panel, which shows the tracks in the kymograph. **Save Tracks** for motor proteins in separate file, one file for each filament is recommended.
6. **Load Tracks** from filament channel as well as drift corrected reference beads from molecule channel. *Note, if filament channel has more than one frame, drift correction as well as averaging for the filaments is necessary.* Choose **Create Offset Map** (Menu→Offset Map) to estimate color offset (**Save Offset Map** file is recommended). To verify color offset activate **Show Offset Map** and use **Align Channels** option. If reference beads are matched incorrectly, delete the respective reference bead and redo **Create Offset Map**.
7. Delete the reference beads by **Select all Molecules** and **Delete selected tracks** (both Right-Click on Molecules Panel). The drift correction and color offset map is saved in the background until it is overwritten or FIESTA is closed.
8. **Load Tracks** of stepping motor proteins (selecting multiple files is possible), make sure only filaments and verified motor tracks are present.
9. **Subtract Drift** to make sure that tracks are drift corrected and choose **Apply Offset Map** to link offset map to the respective tracks. Activate **Align Channels** to calculate the new positions after color correction.
10. **Align Filaments** (Menu→Statistics) in order to align filaments according to the average direction of the motor proteins stepping along them. *The start-point of the filaments will be closer to the attachment position and the end-point closer to the detachment position of the motor proteins.*
11. **Select all Molecules** and choose **Path Statistics** (Menu→Statistics) with the **Filament Centerline** option to project the molecule position on the centerline. The distance along the path (column 3 in PathData field) will then be given as the distance from the start of the filament (real component) as well as for the end of the filament (imaginary component). **Save all Tracks** with motors and filaments for further analysis.

S3. Simulations and data analysis of distributions

Monte-Carlo-Simulation of Poisson stepper

Simulations of motor proteins in stepping assays were performed using a simplified Monte-Carlo-Simulation. In the included example a stepping rate of $k_{\text{step}} = 100\,\text{s}^{-1}$, step size $d_{\text{stepsize}} = 8.3\,\text{nm}$ and detachment rate $k_{\text{off}} = 0.5\,\text{s}^{-1}$ is used. In addition we also account for photobleaching (mix of fluorophores with $k_{\text{bleach}} = 0.125\,\text{s}^{-1}$ and $\rho = 0.5$) and a Schulz distribution ($L_0 = 5\,\mu\text{m}$) was used to get filaments with different lengths for each data set:

$$s(L) = \lambda^2 L e^{-\lambda L} \quad \text{with} \quad \lambda = 1/L_0 \tag{S1}$$

Averaging of the position was included to simulate a frame rate of $f = 10\,\text{s}^{-1}$ (normally used in our experiments) and a positional error (due to tracking uncertainty) is incorporated by adding normal-distributed noise ($\sigma = 20\,\text{nm}$).The complete Monte-Carlo-Simulation can be found in the compressed file and is called `SingleMotorStepping.m`.

Evaluation of FIESTA tracks

The MATLAB code for evaluation of single processive motor proteins tracked using FIESTA can be found in the compressed file and is called `EvaluateTracks.m`. It includes least-square-fitting to obtain the velocities (excluding the first and last frame) and also adjusts for slight mismatches in the time resolution.

Estimation of motility parameters

The MATLAB code for the estimation of the motility parameters can be found in the compressed file. `EstimateMotilityParameters.m` can be used for single exponential distributions and `EstimateMotilityParameters2.m` for double exponential distributions (e.g. different motor populations in the same experiment). The code includes corrections for photobleaching and limited filament length as well as error estimation via bootstrapping.

The MATLAB code includes useful comments on usage as well as on individual sections of the code and an example for the analysis using either a simulation or FIESTA tracks can be found in `ExampleAnalysis.m`. *In order to facilitate the analysis, we incorporated the MATLAB code for evaluation of the tracks and estimation of the motility parameters in FIESTA (version 1.6 or later).*

S4. Velocity estimation

When measuring velocity distribution of motor proteins in stepping assay, statistical tests often show that this distribution is not normal distribution, which is still used to calculate a mean velocity. To investigate the reason behind this discrepancy, we created a Monte-Carlo-Simulation with 10000 traces of motor proteins stochastically stepping with a rate of $k_{\text{step}} = 100\,\text{s}^{-1}$ and step size $d_{\text{step}} = 8\,\text{nm}$ ($k_{\text{off}} = 0$, total time per trace 20 s). We looked at the number of steps N_{steps} taken by each of these simulated motor proteins at specific time points (e.g. $t_1 = 1\,\text{s}$, $t_2 = 3\,\text{s}$, etc). At each time point, N_{steps} is described by a Poisson distribution, which can be approximated with normal distributions ($N_{\text{steps}} > 10$; Figure S1A). Since the velocity is described by $v = N_{\text{steps}} \cdot d_{\text{step}}/T$ the mean velocities are the same at each time point, but the widths of the normal distributions vary (Figure S1B). If the detachment rate is changed to $k_{\text{off}} = 0.5\,\text{s}^{-1}$, each motor has a different interaction time and thereby the velocity distribution of all motors is a mix of normal distributions with the same mean values but different widths. In general, motors with shorter interaction times have a higher variance in the velocity distribution than motors that interact longer (under the same imaging conditions, Figure S1B). Consequently, the observed velocity distribution is not a normal distribution. Here, Figure S2A shows a simulated velocity distribution and the fits (using MLE) of a normal pdf and TLS pdf.

Adding a finite time resolution in the simulation introduced a systematic error when estimating the velocity due to the uncertainty in attachment and detachment. Motor proteins can bind to or unbind from the filament during the acquisition of the frame. Here, the distance will always be underestimated due to time averaging (Figure S3, Table S1). Here, the interaction time will be overestimated because the assumption is that molecules are visible for the complete frame even if they attach or detach within the frame, while the run length is underestimated. For the shown

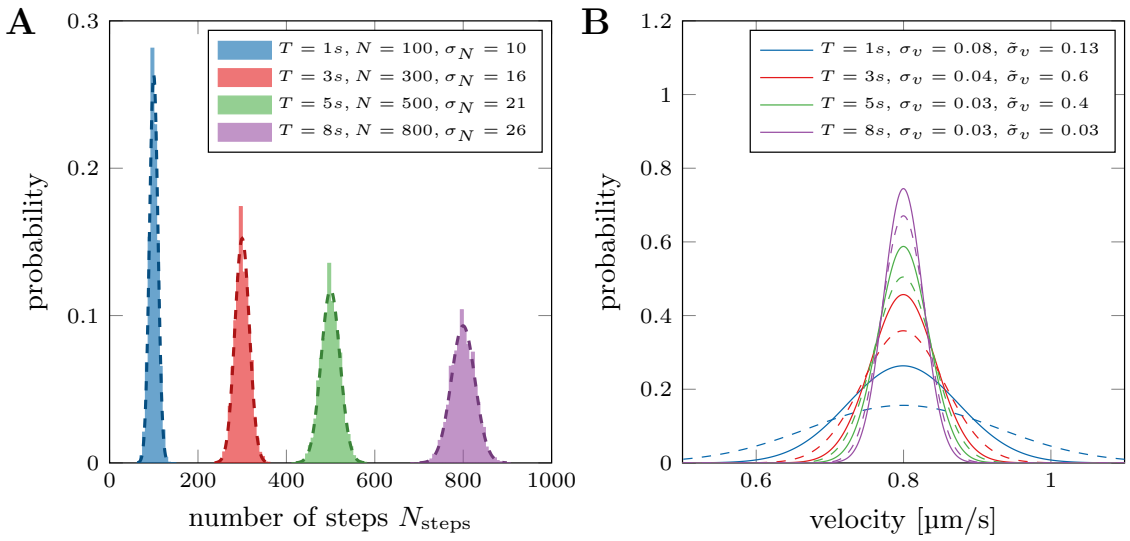

Figure S1 – Poisson distribution for number of steps N_{steps} and resulting velocity distribution: (A) Number of steps N_{steps} taken by each of the simulated motor proteins at specific time points (e.g. $t_1 = 1\,\text{s}$, $t_2 = 3\,\text{s}$, etc; Monte-Carlo-Simulation). The dashed lines show the approximation of the Poisson distributions with normal distributions ($\overline{N}_{\text{steps}} > 10$), each normal distribution yields a different width σ_N. (B) The resulting velocity distribution of the simulated motor proteins at specific time points (e.g. $t_1 = 1\,\text{s}$, $t_2 = 3\,\text{s}$, etc; $v = N_{\text{steps}} \cdot d_{\text{step}}/\tau_{\text{interaction}}$). Solid lines show simulations with infinite frame rate and no added positional error. The dashed lines show simulations with fixed frame rate and added positional error. The mean velocity is the same for each distribution, but the widths σ_v (solid lines, without added positional error) and $\tilde{\sigma}_v$ (dashed lines, with added positional error) are smaller for longer interaction times.

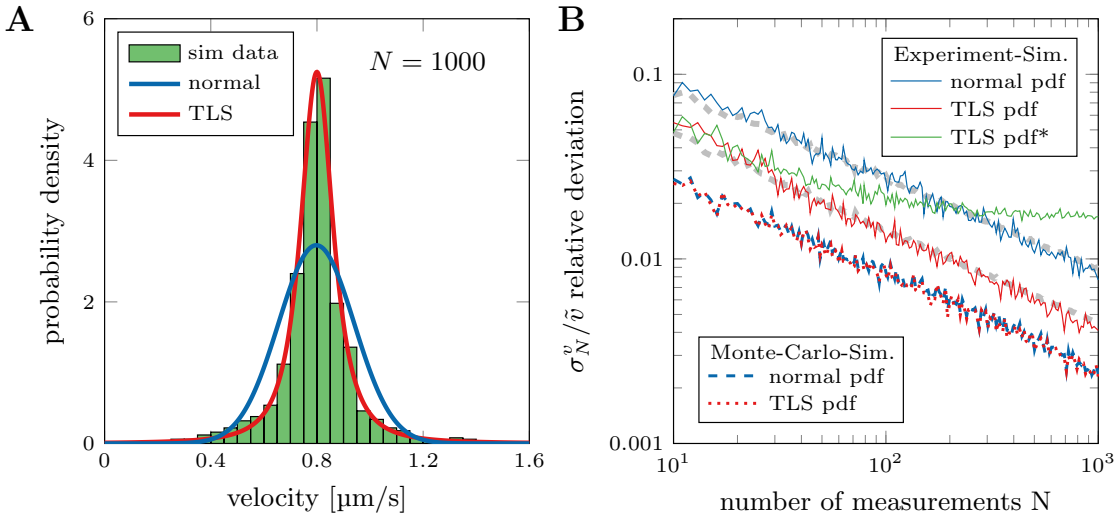

Figure S2 – Evaluation of velocity distributions from simulated traces: (A) Velocity distribution estimated from simulated data (time-averaged, $f = 10\,\mathrm{s}^{-1}$, Monte-Carlo-Simulation, $k_{\mathrm{off}} = 0.5\,\mathrm{s}^{-1}$, with addition positional error, $\sigma = 100\,\mathrm{nm}$). TLS and normal pdf are estimated with MLE. (B) The red/blue dotted/dashed lines show the results using the exact Monte-Carlo simulation whereas the solid lines depict results from simulations that resemble experimental conditions (same data as in Figure 2 in the main text). The green line (TLS*) shows the results when including the first and last point of the trace in the analysis (time-averaged plus positional error; only TLS fitting).

example (Figure S3, Table S1) the precise interaction time of $\tau = 0.93\,\mathrm{s}$ would appear as $\tau = 1.00\,\mathrm{s}$ in the experiment because the motor is seen in all of the 10 frames, while the precise run length $R = 0.73\,\mu\mathrm{m}$ would appear shorter ($R = 0.70\,\mu\mathrm{m}$) in the experiment because the position in the first and last frame is an averaged position. Calculating the velocity $v = R/\tau$ would yield a slower velocity for the time averaged data ($v = 0.70\,\mu\mathrm{m/s}$) than for the precise Monte-Carlo-Simulation ($v = 0.78\,\mu\mathrm{m/s}$). Therefore, the first and last data points of each track should be disregarded from the linear regression when calculating the velocity for each molecule. It will slightly increase the statistical error (due to an apparent shorter interaction time) but no systematic error is introduced in the data analysis (Figure S2B).

	Monte-Carlo-Simulation	time averaged	time averaged + pos. error
run length R	$0.73\,\mu\mathrm{m}$	$0.70\,\mu\mathrm{m}$	$0.74\,\mu\mathrm{m}$
interaction time τ	$0.93\,\mathrm{s}$	$1.00\,\mathrm{s}$	$1.00\,\mathrm{s}$
velocity $v = R/\tau$	$0.78\,\mu\mathrm{m/s}$	$0.70\,\mu\mathrm{m/s}$	$0.74\,\mu\mathrm{m/s}$

Table S1 – Comparison of the motility parameters extracted from the simulated trace shown in Figure S3. The first column displays the results from the Monte-Carlo-Simulation, the second column the results obtained by time averaging the position data and the third column the results obtained using the time averaged data with additional positional error.

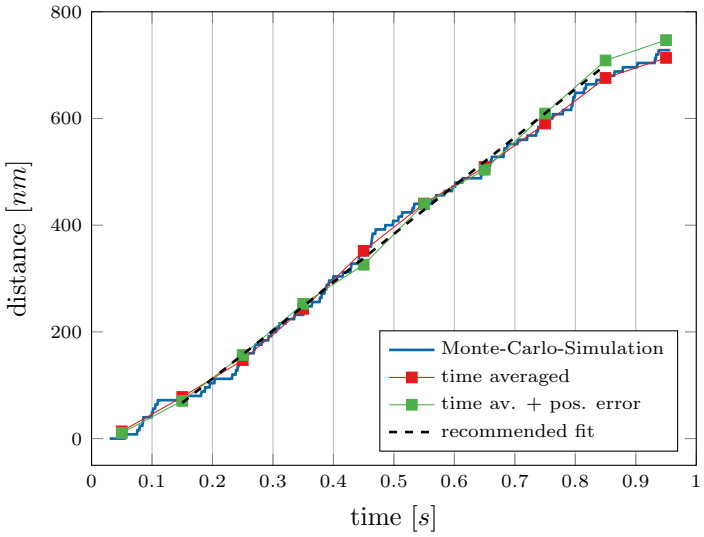

Figure S3 – Simulated trace of a Poisson stepper: The blue line shows the results from the Monte-Carlo-Simulation, whereas the red points show the time averaged trace with one data point per 100 ms (frame rate $f = 10\,\mathrm{s}^{-1}$). The green points include additional positional noise in every data point with $\sigma = 100$ nm. The black dashed line shows the recommended fit of the green data points, excluding the first and last point of the trace since the exact attachment and detachment times or positions are unknown.

S5. Temperature control of the objective

Temperature control of the sample was implemented by using a custom-made hollow brass ring (Figure S4A). The ring (Figure S4B) for the 100x TIRF Objective (Zeiss, Jena, Germany) was manufactured by the Mechanical Workshop at the MPI-CBG (Dresden, Germany) using the design shown in Figure S5. It is connected to a combined cooling and heating unit (F25-MC Refrigerated/Heating Circulator; JULABO GmbH, Seelbach, Germany), via silicone tubing (Figure S4C). When using the TIRF objective, temperature control of the objective is sufficient because the required index-matching oil between the objective and the sample acts as a good heat conductor. Temperature is measured with a small temperature sensor incorporated in the flow channel. The required parts list of the temperature control includes: Combined cooling and heating device (e.g. Julabo Corio or TopTech Series, Thermo Scientific Immersion Circulator, Huber MPC), Hose tail M10, hose tail M5 (e.g. Mafa-Sebald), Silicone tubing (e.g. VWR), Pinch cock (e.g. Bochem, VWR), Flow indicators (e.g. Burkle, SciLabware), Tubing clamps (e.g. Bochem, Burkle).

Unfortunately, small temperature variations can still remain because the temperature of the water bath is controlled. Adding a feedback from the temperature sensor in the flow channel or an additional sensor attached to the objective could further improve the setup. Here, temperature differences due to lack of thermal isolation in the tubing as well as heat generated by the microscope body could be compensated by additional cooling or heating of the water bath especially when trying to achieved temperatures well above or below room temperature.

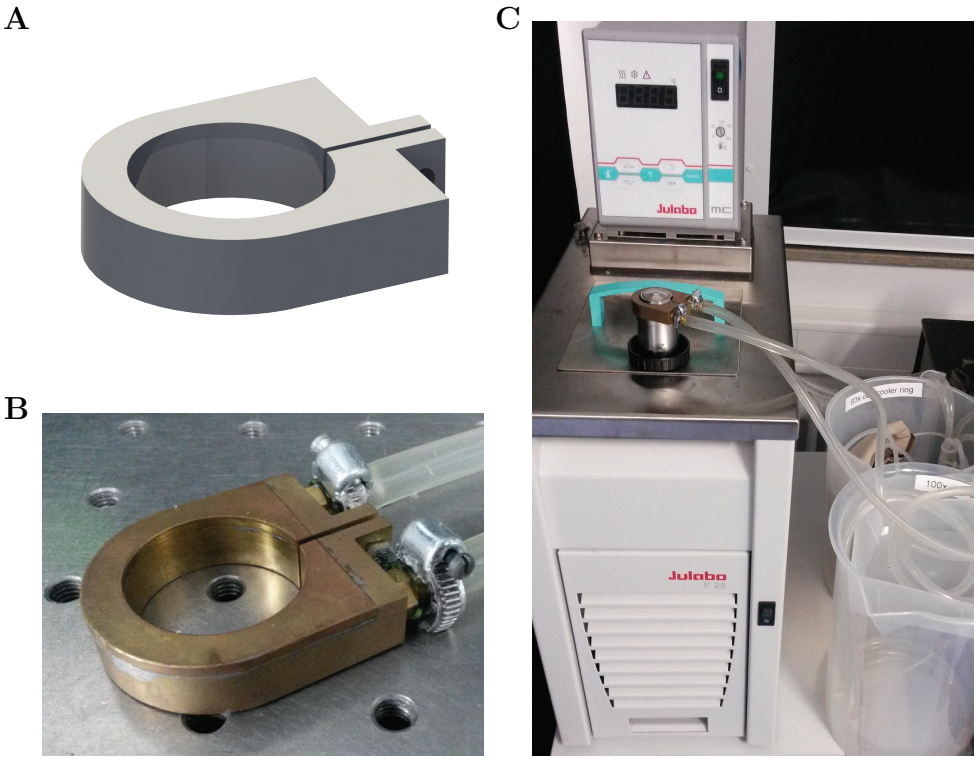

Figure S4 – Temperature control: (A) Design of temperature control ring. (B) Custom-made hollow brass ring for the 100x NA1.46 TIRF Objective (Zeiss, Jena, Germany). (C) Complete temperature control setup with combined cooling heating unit and tubing. Temperature control ring fits on the objective after removing the magnification color code of the objective (rubber band).

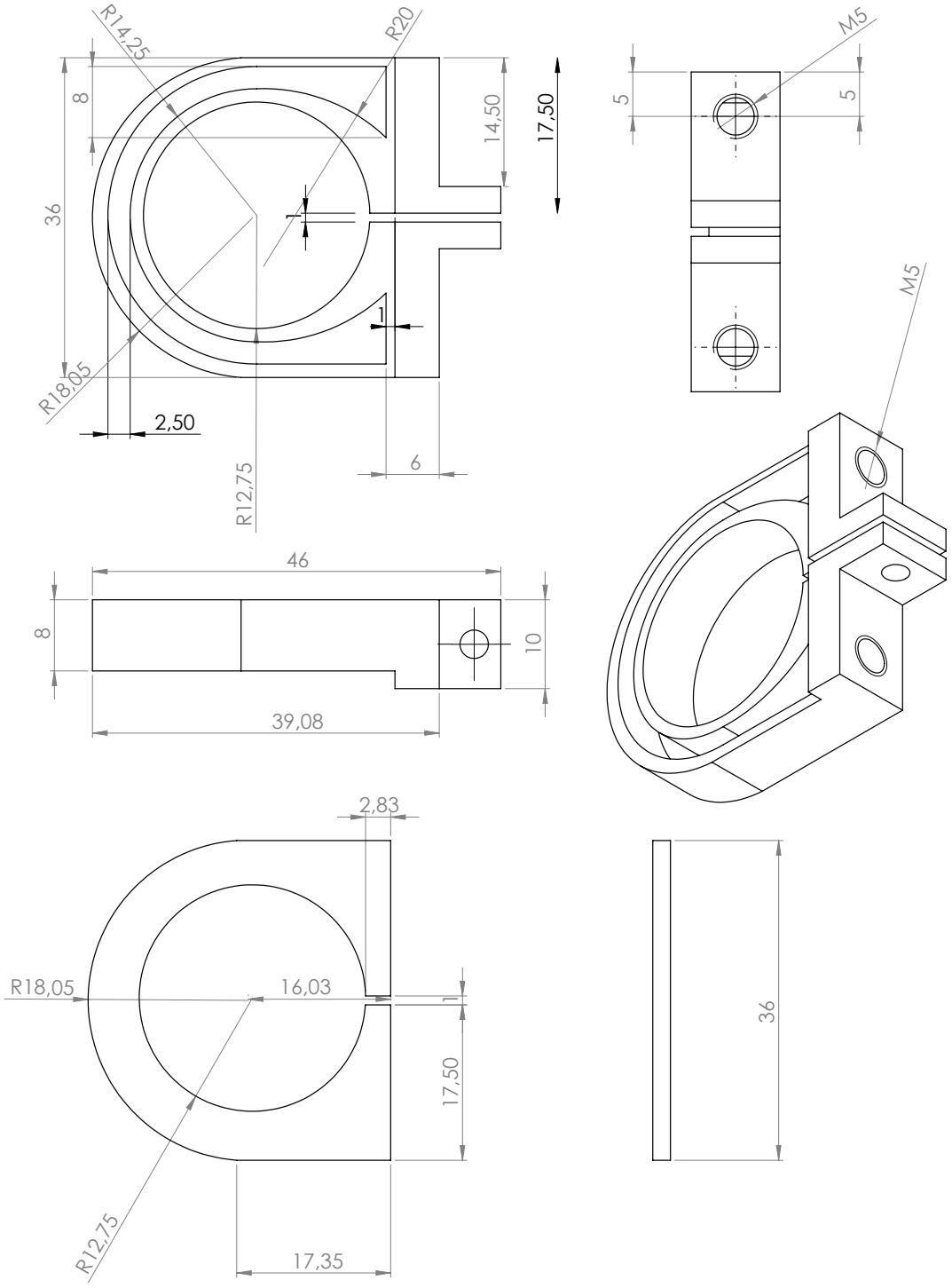

Figure S5 – Technical drawings of temperature control: Hollow brass ring for 100x TIRF Objective (Zeiss, Jena, Germany). Design provided by Friedrich Schwarz (SolidWork files are available on request), custom-made by the Mechachnical Workshop of the MPI-CBG (Dresden, Germany).

S6. Comparison of LSF-CDF(free) and MLE

Comparison of the LSF-CDF(free) and MLE methods shows that both work well with simulated data where the cutoff x_0 is known *a priori*. The mayor disadvantage of the MLE method is the fixed cutoff (specified by the user), which could be hidden within the time resolution or tracking noise. Figure 4B (main text) shows that MLE did not work as well with modified exponential distributions (time resolution $T_{\mathrm{res}} = 0.1\,\mathrm{s}$) when choosing the cutoff $x_0 = 0.5\,\mathrm{s}$ (smallest value), because the distributions also included measurements between $0.45 - 0.5\,\mathrm{s}$. For simulations this can be solved by adjusting the cutoff to $\tilde{x}_0 = x_0 - T_{\mathrm{res}}/2$ (see Figure S6A), but specifying the cutoff with actually experimental data becomes more tricky. Qualitatively the LSF-CDF(free) fitted experimental data better then MLE (using the adjusted cutoff) due to partiality missed events above the cutoff. Nonetheless, both methods can be used to estimate the motility parameters including corrections for censoring (see Figure S6) and it is ultimately a choice of preference.

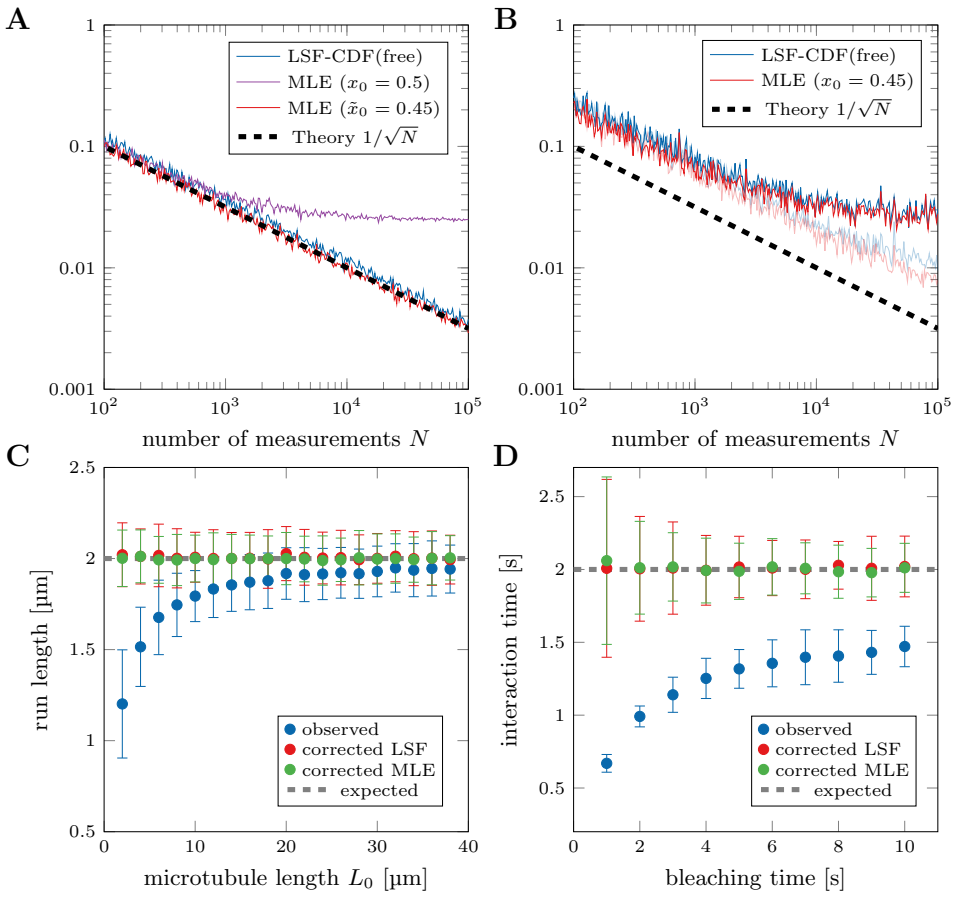

Figure S6 – Comparison of LSF-CDF(free) and MLE: (A) Evaluation of modified exponential distributions (time resolution $T_{\mathrm{res}} = 0.1\,\mathrm{s}$) with LSF-CDF(free) and MLE using a cutoff of $x_0 = 0.5\,\mathrm{s}$. The fixed cutoff in MLE (magenta line) leads to a deviation from the expected behavior (black dotted line). Adjusting the cutoff to $\tilde{x}_0 = x_0 - T_{\mathrm{res}}/2$ shows the MLE has a smaller statistical error than LSF-CDF(free). (B) Evaluation of realistic data including censoring due to limited filament length ($L_0 = 5\,\mu\mathrm{m}$) and photobleaching ($\tau_{\mathrm{bleach}} = 5\,\mathrm{s}$,$N_{\mathrm{bleach}} = 400$). MLE has a smaller statistical error than LSF-CDF(free) and the statistical error of bleaching parameters leads to the deviation at large N. Increasing the number of measurements for the bleaching evaluation reduces this deviation (lighter lines $N_{\mathrm{bleach}} = 10000$). (C) Simulation of motor proteins censored due to limited filament length (same data as Figure 5B in the main text) analyzed with LSF-CDF(free) and MLE for comparison. (D) Simulations with finite filament lengths and photobleaching of motor proteins (same data as Figure 6B in the main text) analyzed with LSF-CDF(free) and MLE for comparison.

S7. Estimation of interaction time and run length

When evaluating exponential distributions using least-squares-fitting of the cumulative probability distribution function (LSF-CDF) the resolution of the measurements becomes important. Even though the cdf can be described by equation S2, the exact point for least-squares-fitting will not be identical with the value of the measurement x_i^c. Here, Figure S7A shows the cdf of a complete distribution with (red line) and without (blue line) a finite time resolution of 1 s (exaggerated to illustrate the problem). The corresponding data points needed for the LSF-CDF are in the middle of steps that are introduced by the chosen resolution. Therefore, the cdf needs to be adjusted by calculating the middle of the steps before fitting (when using `ecdf` (MATLAB) the edges of the 'stairs' are calculated).

$$y_c = f(x \mid \mu) = 1 - e^{\frac{-(x-x_0)}{\mu}} \qquad x_i^c = v_i \quad \forall i = 0, 1, \ldots; \quad y_i^c = \frac{\sum_j 1}{N} \quad \forall j : v_j < v_i \tag{S2}$$

Additionally, the exact time points for image acquisition may not be correlating with the time resolution (e.g. MetaMorph time-stamps frequently show deviations of < 1 ms). Therefore, traces with 5 frames might show up with interaction times between 500.0 and 500.5 ms. This would shift some data points from the middle of the steps toward the edges. So instead of using the exact time stamp, we calculate the interaction time by multiplying the number of frames by the chosen time resolution.

We also investigated how a fixed cutoff value x_0 in equation S2 could influence the result. Here, especially if the interaction time is evaluated (with specific time resolution), a fixed x_0 could introduce systematic errors. Figure S7B shows a simulated cdf with $x_i \geq 3$ s (green line) and a simulated cdf with $x_i \geq 2.5$ s (blue line), both without time resolution. In addition, an observed cdf with time resolution of 1 s is shown (red line) with $x_i \geq 3$ s. Here, using a cutoff value $x_0 = 3$ s would introduce a systematic error because the observed cdf also includes measurements $2.5 < x_i < 3$ s. Therefore, rather than fixing x_0 as a constant (e.g. $x_0 = 2.5$ s in our example), it should be included as a fit parameter in the LSF-CDF so that the shape of the exponential distribution is estimated without forcing the cdf through a particular point.

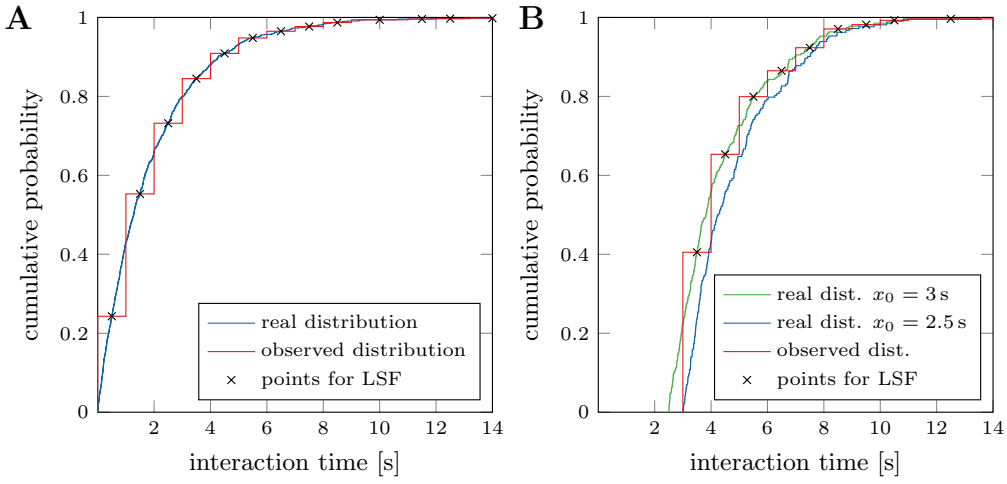

Figure S7 – Least-squares-fitting of cumulative probability distributions: (A) Cumulative probability distribution (blue line) of simulated exponential distribution ($\mu = 2$, $N = 1000$). Same distribution with resolution of 1 is shown in red. Points for least-squares-fitting are shown with a black x. (B) Simulated distributions with cutoff $x_0 = 3$ s (green line) and $x_0 = 2.5$ s (blue line), both without time resolution. The observed distribution (with time resolution of 1 s) with cutoff $x_0 = 3$ s (after adjusting for finite time resolution) is shown in red. The observed distribution can be best explained by green distribution (cutoff at 2.5 s). Points for least-square fitting are shown with a black x.

S8. Kaplan-Meier estimator

The Kaplan-Meier estimator was historically used in medical studies to to estimate survival rates of patients after treatment. The non-parametric statistic can be used to calculate the survival function from lifetime data even if the data is not complete or censored. Nowadays, its application is also found in other fields like economics, engineering or agriculture. In order to illustrate the application of the Kaplan-Meier estimator to correct for censored data points (due to limitations of the study or the experiment), the following analogies can be used to compare medical studies with the analysis of the motility parameters in single motor protein stepping assay:

death of patient is not related to the treated disease (e.g. car accident)	$\Leftrightarrow$	motor protein detaches because it reached the filament end
patient did not die during the study	$\Leftrightarrow$	motor protein is still moving along the filament when acquisition of the image stack is terminated
patient leaves the study with unknown reason (e.g. due to relocation)	$\Leftrightarrow$	observation of the motor protein is no longer possible after photobleaching of the fluorophore

Since the observed lifetimes, or in this case interaction times, are censored due to the experimental limitations, the Kaplan-Meier estimator can be used to calculate the real survival function, which can then be transformed into the cumulative probability density. Table S2 illustrates the method using a concrete randomly generated example from a exponential distribution using a interaction time $\tilde{\tau} = 5\,\mathrm{s}$ (rounded values for simplification). From the original data set two values (7 & 8) are reduced and treated as censored data points. Here, while estimating τ from the original data set yields $\tau = 4.9$, analysis of the observed (and censored) data set would result in $\tau = 4.0$. For this

original data set: 1,1,2,3,4,5,5,7,8,13 $\quad \tau = 4.9$
observed data set: 1,1,2,3,4,5,5,4,2,13 $\quad \tau = 4.0$
(red values are censored)

t	d	r	$S(t_i)_{\mathrm{observed}}$	$C(t_i)$	d	c	r	$S(t_i)_{\mathrm{corrected}}$	$C(t_i)$
1	2	10	$1 \cdot (1 - 2/10) = 0.8$	0.2	2	0	10	$1 \cdot (1 - 2/10) = 0.8$	0.2
2	2	8	$0.8 \cdot (1 - 2/8) = 0.6$	0.4	1	1	8	$0.8 \cdot (1 - 1/8) = 0.7$	0.3
3	1	6	$0.6 \cdot (1 - 1/6) = 0.5$	0.5	1	0	6	$0.7 \cdot (1 - 1/6) = 0.58$	0.42
4	2	5	$0.5 \cdot (1 - 2/5) = 0.3$	0.7	1	1	5	$0.58 \cdot (1 - 1/5) = 0.46$	0.54
5	2	3	$0.3 \cdot (1 - 2/3) = 0.1$	0.9	2	0	3	$0.46 \cdot (1 - 2/3) = 0.15$	0.85
13	1	1	$0.9 \cdot (1 - 1/1) = 0$	1	1	0	1	$0.15 \cdot (1 - 1/1) = 0$	1

analysis of the data sets with bootstrapping ($n = 1000$)
original data set: $\tau_{\mathrm{b}} = 4.9 \quad \sigma_{\mathrm{b}} = 1.1$
observed data set: $\tau_{\mathrm{b}} = 4.0 \quad \sigma_{\mathrm{b}} = 1.1$
corrected data set: $\tau_{\mathrm{b}} = 5.1 \quad \sigma_{\mathrm{b}} = 1.5$

Table S2 – Example for the application of the Kaplan-Meier estimator: d_i denotes the number of motors with a certain interaction time t_i and r_i the number of attached motors with an interaction time $t \geq t_i$. The hazard rate $h(t_i) = d_i/r_i$ can then be used to calculate the survival probability $S(t_i) = S(t_{i-1}) \cdot (1 - h(t_i))$ (with $S(t_0) = 1$). The cumulative probability C (probability of detachment) can then be estimated using $C(t_i) = 1 - S(t_i)$. When using the Kaplan-Meier estimator to correct the survival probability S censored events are not counted in d_i but are included in r_i, which results in an adjusted cumulative probability C.

reason, correction of the analysis using the Kaplan-Meier estimator is necessary and only requires information on which measurements are censored.

For analysis of the interaction time survival analysis can be used. Briefly, let d_i be the number of motors after a certain interaction time t_i and r_i the number of attached motors with an interaction time $t \geq t_i$. The hazard rate $h(t_i) = d_i/r_i$ can then be used to calculate the survival probability $S(t_i) = S(t_{i-1}) \cdot (1 - h(t_i))$ (with $S(t_0) = 1$). The cumulative probability C (probability of detachment) can then be estimated using $C(t_i) = 1 - S(t_i)$ for each unique interaction time measurement in the data set (Table S2). When using the Kaplan-Meier estimator to correct the survival probability S censored events are not counted in d_i but are included in r_i, which also results in an adjusted cumulative probability C. Evaluating the cumulative probability using the exponential distribution now yields a corrected interaction time. This analysis can be easily combined with the bootstrapping method to estimate the statistical error and analysis of the generated example, yielding $\tau = 4.9 \pm 1.1(\sigma_\mathrm{b})$ for the original data set, $\tau = 4.0 \pm 1.1(\sigma_\mathrm{b})$ for the observed data set and $\tau = 5.1 \pm 1.5(\sigma_\mathrm{b})$ for the observed data set when applying the Kaplan-Meier estimator. In the last result, the systematic error of the observed data set (due to censoring) is reduced at the cost of increasing the statistical error.

S9. Alternative method for verification of length correction

Figure 5C in the main text shows the proof-of-principle for the proposed filament length correction. An alternative method to verify the length correction is to reduce the data set to only include traces that are not influenced by the filament length. Here, the longest trace in the complete data set ($N = 5208$) was 11.4 µm and therefore we only looked at molecules that landed more than 10 µm away from the filament end. None of the remaining traces ($N = 914$) were now classified as end events, but some were still censored either by the edge of the field of view or the beginning/end of imaging. Table S3 shows that the length-corrected result of the complete data set is the same as the observed result of the reduced data set. Unfortunately, the number of measurements is greatly reduced which drastically increases the statistical error. Therefore, the proposed length correction method using the complete data set is more precise for estimating the motility parameters.

data set	complete	reduced
N_total	5208	914
$N_\mathrm{end-event}$	480	0
$N_\mathrm{misc-censored}$	112	13
R_observed	0.61 ± 0.03µm	0.67 ± 0.06µm
$R_\mathrm{corrected}$	0.68 ± 0.03µm	0.67 ± 0.06µm

Table S3 – Comparison of the run lengths using the complete and reduced data set (landing position more than 10 µm away from the filament end). The reduced data set does not include any end-events and is not influenced by the filament length, but also only uses less than a fifth of the original measurements, which roughly doubles the statistical error (errors are given as $\Delta R = 2 \cdot \sigma_R$).

S10. Evaluation of photobleaching

Photobleaching correction requires precise information about the bleaching behavior of the fluorophores. Here, we characterize the photobleaching behavior of motor proteins labeled with one or two fluorophores. Whereas, photobleaching of one fluorophore is described by a single exponential, the bleaching distribution changes from a single exponential to a superposition of two exponential functions for two fluorophores (Figure S8A). The bleaching probabilities (to total darkness) can then be described by:

$$\text{One fluorophore:} \quad P = k_{\text{bleach}} e^{-k_{\text{bleach}} x} \tag{S3}$$

$$\text{Two fluorophores:} \quad P = 2k_{\text{bleach}}(e^{-k_{\text{bleach}} x} - e^{-2k_{\text{bleach}} x}) \tag{S4}$$

$$\text{Mix:} \quad P = k_{\text{bleach}}(2-\rho)e^{-k_{\text{bleach}} x} + 2k_{\text{bleach}}(\rho-1)e^{-2k_{\text{bleach}} x} \tag{S5}$$

Since not all fluorophores are active, the observed distribution will yield a mix of one and two fluorophore bleaching (Figure S8B). While it can be described analytically, the combination with the detachment rate of the motors is non-trival. Additionally, the experimental limitations, especially the short missing events, make it almost impossible to obtain a stable solution with LSF-CDF(free). In fact, in most cases it was not possible to get the true values when using simulations with a mixed bleaching distribution. Therefore, we opted to analyze the bleaching times of the individual fluorophores and measure the ratio between one and two active fluorophores (see S11).

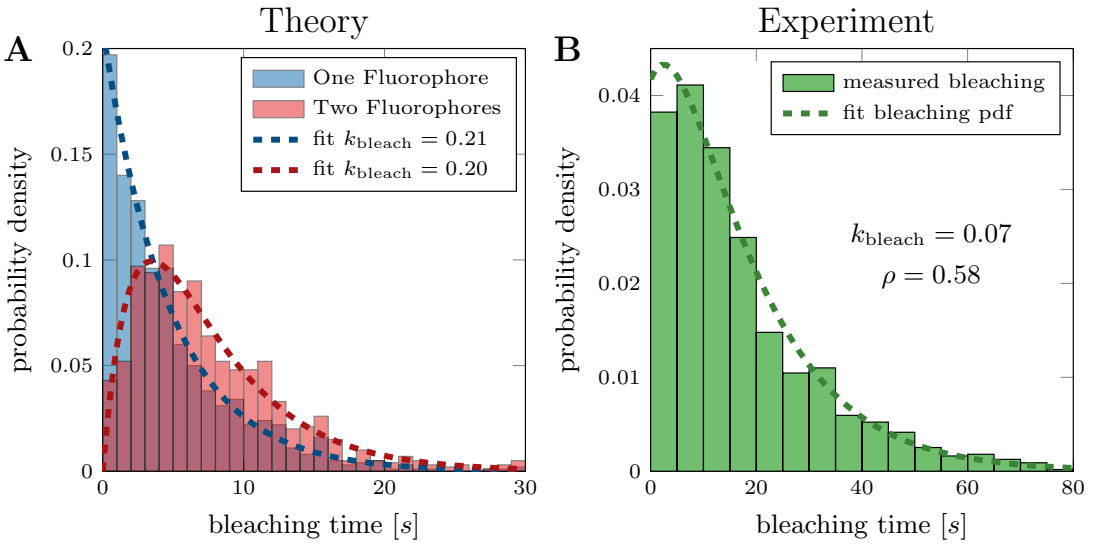

Figure S8 – Bleaching correction: (A) pdf of bleaching times for molecules with one or two active fluorophores ($\tau_{\text{bleach}} = 5s, N = 1000$). Bleaching of one fluorophore can be described with an exponential function. Bleaching times of two fluorophores are logical disjunction of two exponential functions (molecules disappear when the last of the two fluorophores bleaches). (B) Measured bleaching times of immobilized molecules disregarding one and two step bleaching. The resulting distribution is a mixture of the distributions shown in Figure S8A.

S11. Measuring photobleaching distribution with FIESTA

FIESTA (version 1.05.0005 or later) can be used to extract the bleaching times for immobilized fluorescently-labeled motor proteins semi-automatically. Here, the tracked positions of the immobilized motors in all frames are mapped and categorized in clusters. Then, the intensity of random clusters is displayed over time and the user can select bleaching steps:

1. **Open Stack** (Menu→Data) and track immobilized molecules (set verification steps to 0 to disable connecting tracks).
2. **Load Objects** (Menu→Data) from the results file.
3. Choose **Bleaching Evaluation** (Menu→Statistics), enter minimum number of necessary objects per cluster and choose box size for intensity integration in pixels. FIESTA will display a window with the integrated intensity of a box around a cluster of tracked objects (minimum cluster size set by user) over time.
4. Choose visible bleaching steps by clicking at the position in the graph (either one or two step bleaching). If no bleaching step is visible or the intensity shows other fluctuation skip the molecule.
5. Save results and use the file in the run length or interaction time evaluation.