

Biophysical Journal, Volume 113

Supplemental Information

**Challenges in Estimating the Motility Parameters of Single Processive
Motor Proteins**

Felix Ruhnaw, Linda Kloß, and Stefan Diez

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

F. Ruhnnow, L. Kloß, S. Diez, Felix Ruhnnow^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

S1	Experimental considerations	2
S2	Tracking with FIESTA	3
S3	Simulations and data analysis of distributions	4
S4	Velocity estimation	5
S5	Temperature control of the objective	8
S6	Comparison of LSF-CDF(free) and MLE	10
S7	Estimation of interaction time and run length	11
S8	Kaplan-Meier estimator	12
S9	Alternative method for verification of length correction	13
S10	Evaluation of photobleaching	14
S11	Measuring photobleaching distribution with FIESTA	15

*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:

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

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

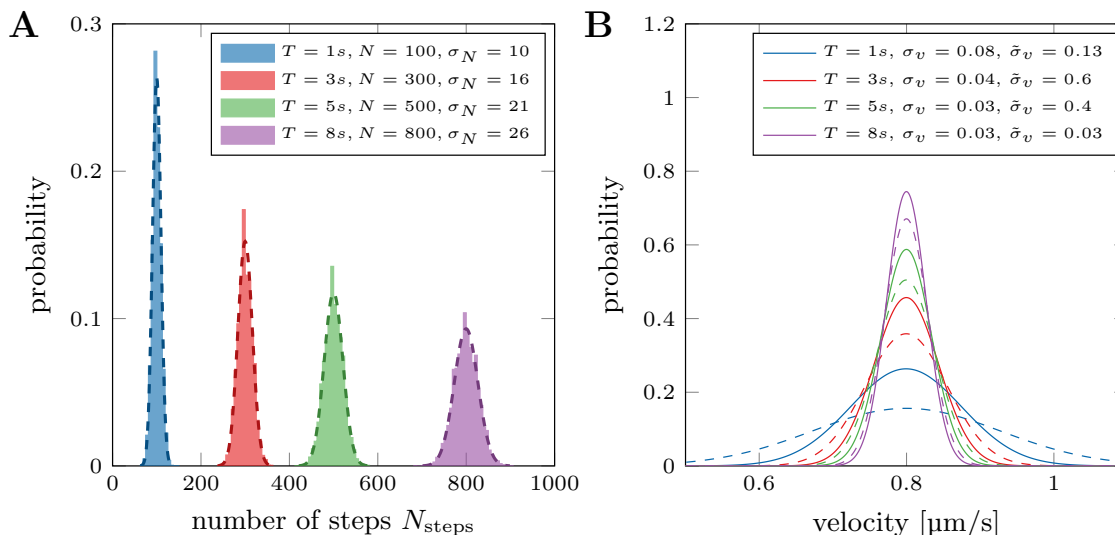


Figure S1 – Poisson distribution for number of steps N_{steps} and resulting velocity distribution: (A) Number of steps N_{steps} taken by each of the simulated motor proteins at specific time points (e.g. $t_1 = 1 \text{ s}$, $t_2 = 3 \text{ s}$, etc; Monte-Carlo-Simulation). The dashed lines show the approximation of the Poisson distributions with normal distributions ($N_{\text{steps}} > 10$), 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.

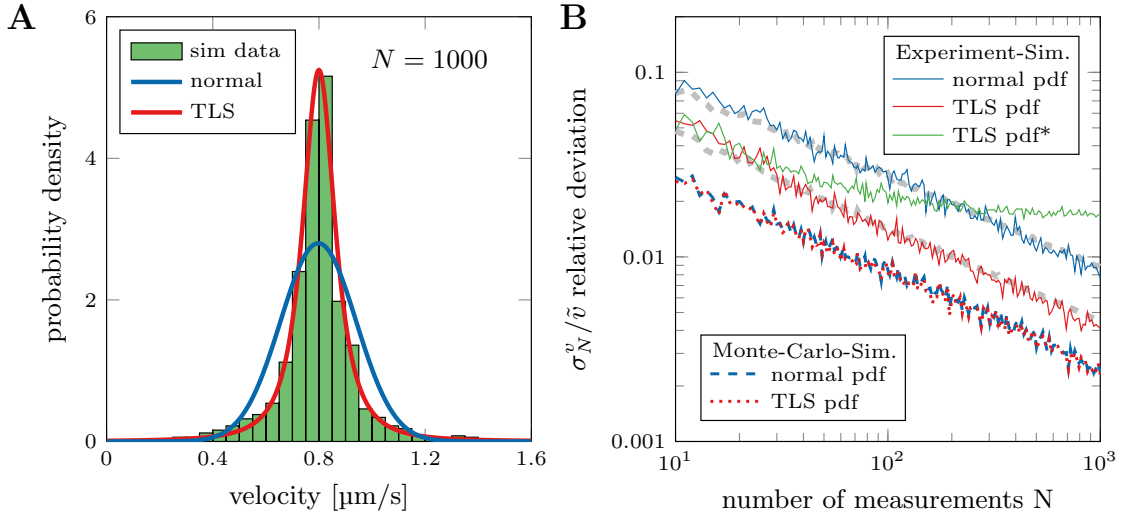


Figure S2 – Evaluation of velocity distributions from simulated traces: (A) Velocity distribution estimated from simulated data (time-averaged, $f = 10 \text{ s}^{-1}$, Monte-Carlo-Simulation, $k_{\text{off}} = 0.5 \text{ s}^{-1}$, with addition positional error, $\sigma = 100 \text{ nm}$). TLS and normal 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 $\tau = 0.93 \text{ s}$ would appear as $\tau = 1.00 \text{ s}$ in the experiment because the motor is seen in all of the 10 frames, while the precise run length $R = 0.73 \mu\text{m}$ would appear shorter ($R = 0.70 \mu\text{m}$) 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 ($v = 0.70 \mu\text{m/s}$) than for the precise Monte-Carlo-Simulation ($v = 0.78 \mu\text{m/s}$). 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	$0.73 \mu\text{m}$	$0.70 \mu\text{m}$	$0.74 \mu\text{m}$
interaction time τ	0.93 s	1.00 s	1.00 s
velocity $v = R/\tau$	$0.78 \mu\text{m/s}$	$0.70 \mu\text{m/s}$	$0.74 \mu\text{m/s}$

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.

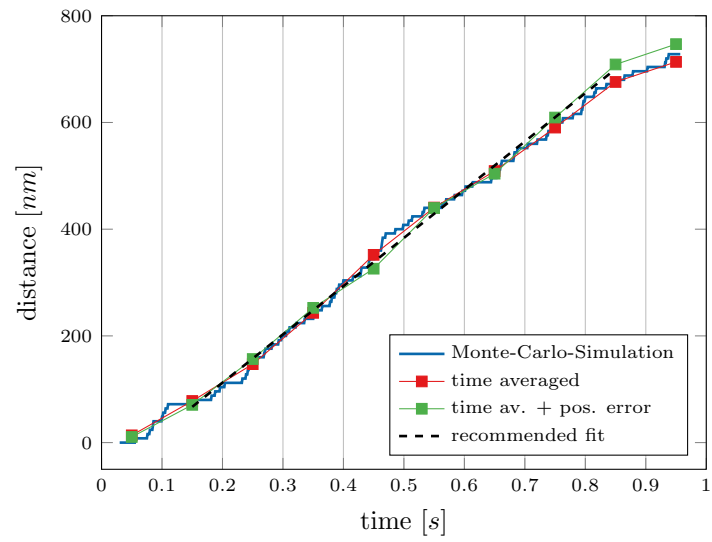


Figure S3 – Simulated trace of a Poisson stepper: The blue line shows the results from the Monte-Carlo-Simulation, whereas the red points show the time averaged trace with one data point per 100 ms (frame rate $f = 10 \text{ s}^{-1}$). The green points include additional positional noise in every data point with $\sigma = 100 \text{ nm}$. 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.

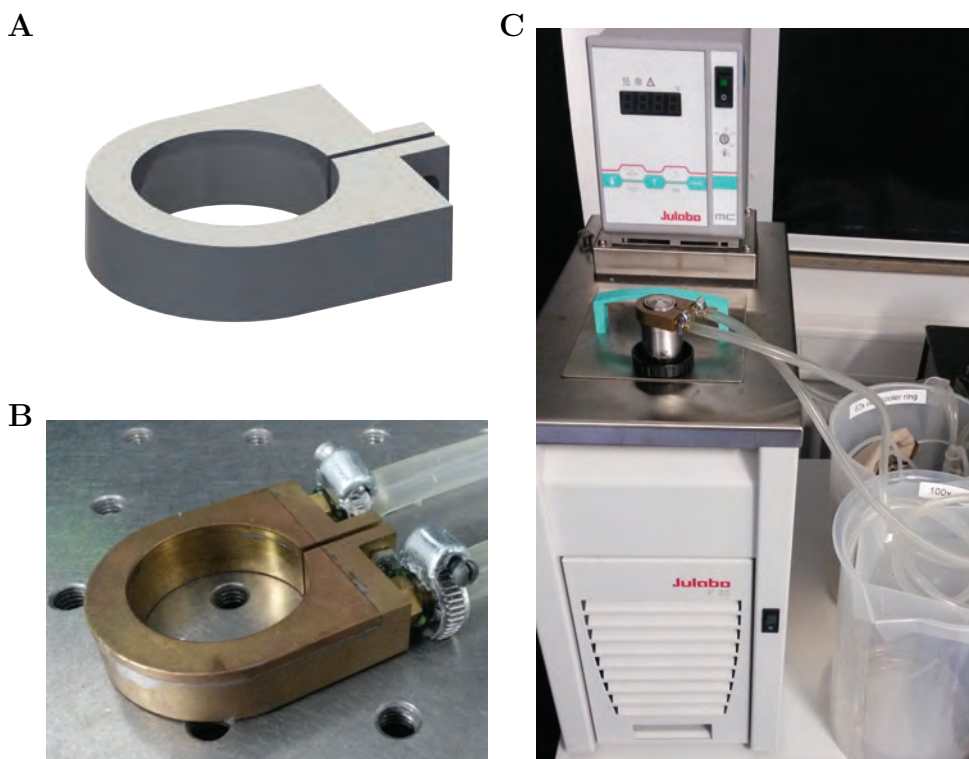


Figure S4 – Temperature control: (A) Design of temperature control ring. (B) Custom-made hollow brass ring for the 100x NA1.46 TIRF Objective (Zeiss, Jena, Germany). (C) Complete temperature control setup with combined cooling heating unit and tubing. Temperature control ring fits on the objective after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

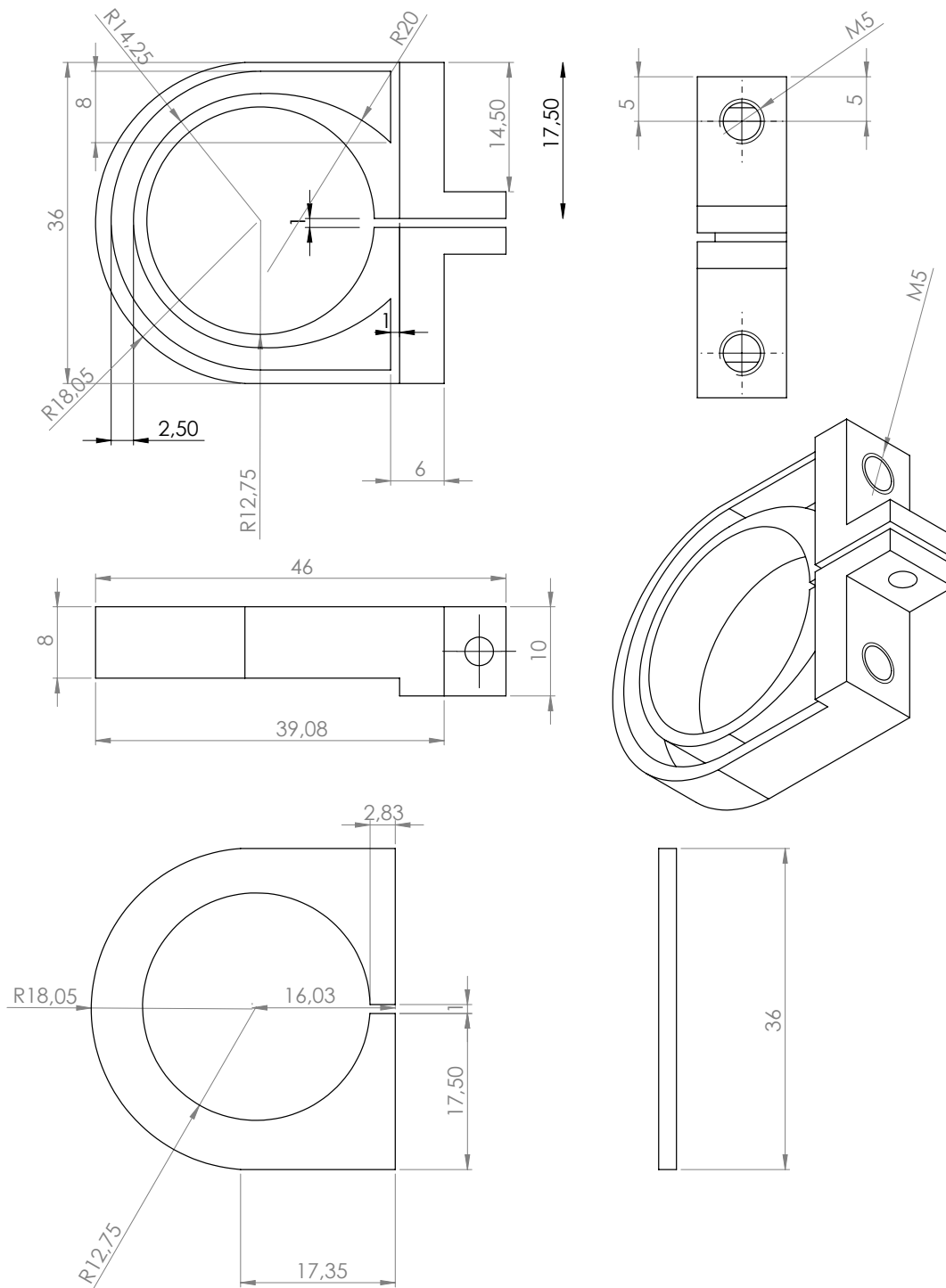


Figure S5 – Technical drawings of temperature control: Hollow brass ring for 100x TIRF Objective (Zeiss, Jena, Germany). Design provided by Friedrich Schwarz (SolidWork files are available on request), custom-made by the Mechanical Workshop of the MPI-CBG (Dresden, Germany).

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 $T_{\text{res}} = 0.1$ s) when choosing the cutoff $x_0 = 0.5$ s (smallest value), because the distributions also included measurements between $0.45 - 0.5$ s. For simulations this can be solved by adjusting the cutoff to $\tilde{x}_0 = x_0 - T_{\text{res}}/2$ (see Figure S6A), but specifying the cutoff with actually experimental data becomes more tricky. Qualitatively the LSF-CDF(free) fitted experimental data better than 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.

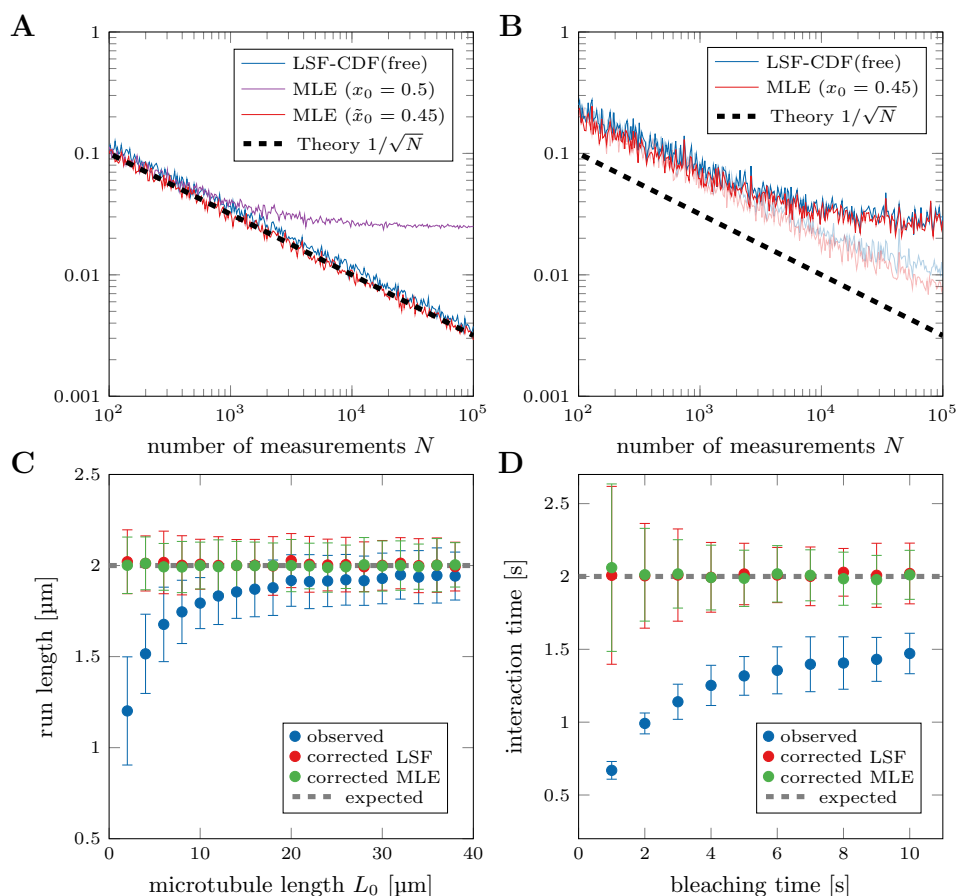


Figure S6 – Comparison of LSF-CDF(free) and MLE: (A) Evaluation of modified exponential distributions (time resolution $T_{\text{res}} = 0.1$ s) with LSF-CDF(free) and MLE using a cutoff of $x_0 = 0.5$ s. The fixed cutoff in MLE (magenta line) leads to a deviation from the expected behavior (black dotted line). Adjusting the cutoff to $\tilde{x}_0 = x_0 - T_{\text{res}}/2$ shows the MLE has a smaller statistical error than LSF-CDF(free). (B) Evaluation of realistic data including censoring due to limited filament length ($L_0 = 5 \mu\text{m}$) and photobleaching ($\tau_{\text{bleach}} = 5$ s, $N_{\text{bleach}} = 400$). 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 $N_{\text{bleach}} = 10000$). (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).

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

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.

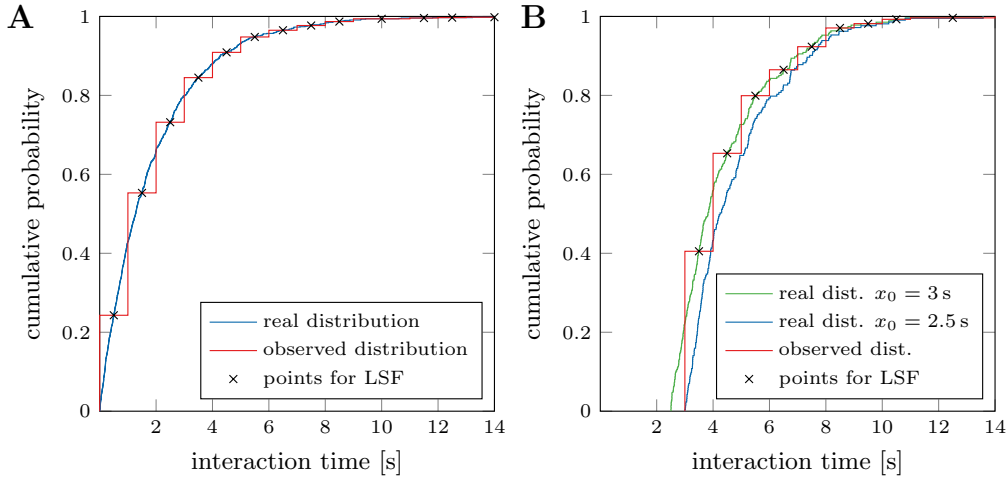


Figure S7 – Least-squares-fitting of cumulative probability distributions: (A) Cumulative probability distribution (blue line) of simulated exponential distribution ($\mu = 2$, $N = 1000$). Same distribution with resolution of 1 is shown in red. Points for least-squares-fitting are shown with a black 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 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 $\tilde{\tau} = 5$ 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	$\tau = 4.9$
	observed data set: (red values are censored)	1,1,2,3,4,5,5,4,2,13	$\tau = 4.0$

t	d	r	$S(t_i)_{\text{observed}}$	$C(t_i)$	d	c	r	$S(t_i)_{\text{corrected}}$	$C(t_i)$
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:	$\tau_b = 4.9 \quad \sigma_b = 1.1$
observed data set:	$\tau_b = 4.0 \quad \sigma_b = 1.1$
corrected data set:	$\tau_b = 5.1 \quad \sigma_b = 1.5$

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 $\tau = 4.9 \pm 1.1(\sigma_b)$ for the original data set, $\tau = 4.0 \pm 1.1(\sigma_b)$ for the observed data set and $\tau = 5.1 \pm 1.5(\sigma_b)$ 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 $11.4 \mu\text{m}$ and therefore we only looked at molecules that landed more than $10 \mu\text{m}$ 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
$N_{\text{end-event}}$	480	0
$N_{\text{misc-censored}}$	112	13
R_{observed}	$0.61 \pm 0.03 \mu\text{m}$	$0.67 \pm 0.06 \mu\text{m}$
$R_{\text{corrected}}$	$0.68 \pm 0.03 \mu\text{m}$	$0.67 \pm 0.06 \mu\text{m}$

Table S3 – Comparison of the run lengths using the complete and reduced data set (landing position more than $10 \mu\text{m}$ 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:

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

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

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

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-trivial. 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).

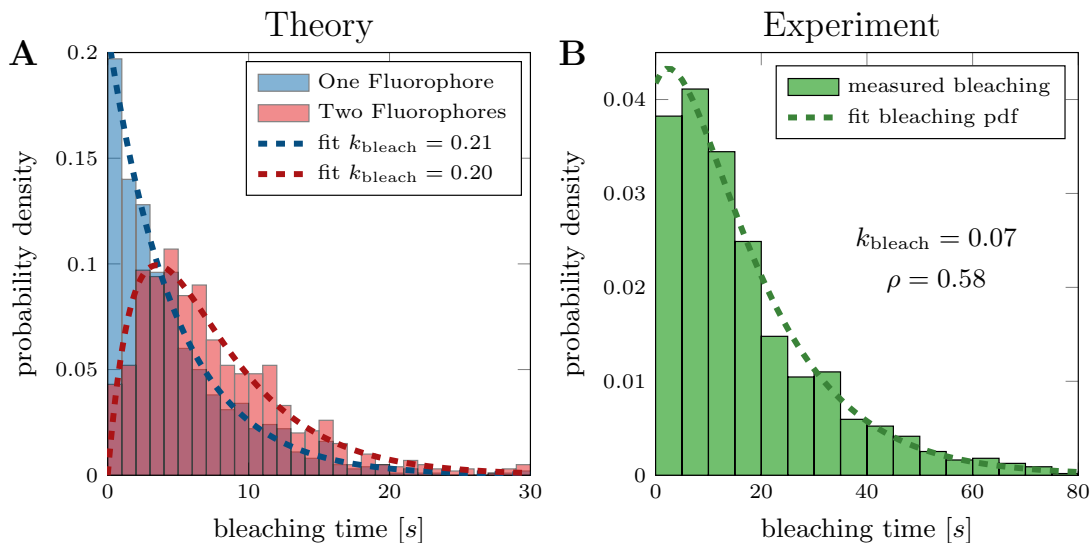


Figure S8 – Bleaching correction: (A) pdf of bleaching times for molecules with one or two active fluorophores ($\tau_{\text{bleach}} = 5\text{s}, N = 1000$). Bleaching of one fluorophore can be described with an exponential function. Bleaching times of two fluorophores are logical disjunction of two 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.