6

A fast and automated step detection method for analysing single-molecule trajectories

In preparation .

Luuk Loeff*, Jacob Kerssemakers*, Chirlmin Joo** & Cees Dekker**

* These authors have contributed equally to this work

** Co-corresponding authors

Kavli Institute of NanoScience and Department of BioNanoScience, Delft University of Technology, 2628 CJ, Delft, The Netherlands.

6.1 Abstract

Single-molecule techniques have made it possible to study the molecular dynamics of nucleic acids and proteins with a high spatial and temporal resolution. Accurate determination of the states in single-molecule events provides value information about the inherent kinetic properties of biomolecules. Here we present a fast and automated step detection method that is capable of detecting steps in large datasets without any prior knowledge on the distribution of step sizes or their location. Step detection is based on a series of partition events that minimize the variance between fit and the data (chi-squared). After each step fit, the quality of the fit is assessed by performing a secondary fit on the data. A multi-pass strategy that determines the optimal fit for the data over two rounds allows *Stepfinder* to automatically detect steps. The user-friendly interface and the automated step detection of the enhanced *Stepfinder* algorithm provides a robust "hands-off" fitting procedure that can be executed by anyone without programming knowledge in less than 10 minutes.

6.2 Introduction

Over the last two decades, fluorescence based single-molecule techniques have greatly enhanced our understanding of complex biological processes [1–5]. These techniques have made it possible to track the molecular dynamics of single proteins and protein complexes with a (sub)nanometer spatial resolution and a (sub)millisecond timescale [1, 2, 4]. Single-molecule fluorescence techniques have been used to determine the stoichiometry, binding kinetics and conformational dynamics of both nucleic acids and proteins [6–11]. Force spectroscopy (e.g. optical or magnetic tweezers) has been exploited as a versatile tool for probing the forces and motions that are associated with biological molecules [4, 12–14]. Lastly, nanopores have provided a powerful tool for label-free detection of nucleic acids and proteins [15–18].

Accurate determination of the states in single-molecule events provides valuable information about the kinetic properties that drive the biological function of proteins. The states in single-molecule fluorescence events often have to be analysed by a trained person, picking out each state manually. This manual analysis poses several drawbacks: (i) manual analysis relies on one's experience in distinguishing background noise from legitimate state-to-state transitions; (ii) it may be subject to a user bias; (iii) short events are likely to be missed; (iv) manually picking out events is a time-consuming process. These shortcomings affect both the reliability and reproducibility of data analysis and thereby manual analysis becomes almost impossible when the data exhibits a large number of states (e.g. more than three).

To reliably and reproducibly analyse data, several automated step detection algorithms have been developed over the past two decades [19–22]. Commonly used approaches for automated step detection rely on thresholding [21] or pairwise distribution analysis [23, 24], which do not suffice when the data exhibits more than three distinct steps of variable size. Another commonly used approach is based on Hidden Markov Modelling (HMM) [19], that requires specification of the number of different states that are visited during the time course of an experiment, which is commonly a unknown variable when doing experiments [25]. To work around this limitation, HMM can be combined with Bayesian nonparametrics [26], allowing one to use HMM without any knowledge on the number of visited states a priori [22, 27, 28]. However, HMM assumes that each state is visited successively, making the algorithm only suitable for events that occur frequently.

We previously reported on a step finding algorithm (*Stepfinder*) that is based on chi-squared minimization, capable of detecting steps in step trains without any prior knowledge on their size or location [20]. After its release, the chi-squared minimization algorithm gained a great interest in the field of biophysics [29] and has been applied on the analysis of trajectories of a wide variety of techniques. These techniques include: optical and magnetic tweezers, single-molecule FRET and nanopores (Figure 6.1) [20, 30–36]. Despite its popularity, the algorithm faced several caveats: (i) the algorithm was subject to user bias, requiring the user to determine final the number of steps (ii) the algorithm was computationally demanding when presented with large datasets (iii) step evaluation failed when presented with data that exhibited a large variety of step-sizes, which especially holds true for baseline type trajectories (iv) the algorithm lacked a user-friendly interface.

Here, we present an enhanced version of *Stepfinder* that allows for high-throughput and automated step detection (Chapter 3 on page 73). We created a user-friendly interface that simplifies step detection in single-molecule trajectories. First, we recapitulate how the step finding procedure is performed and how the optimal number of fitted steps is determined. Next, we elaborate on how the selection criteria for the optimal number of steps change when data exhibits a wide variety of step-sizes and plateau lengths. We show that these considerations lead to a robust "hands-off" fitting procedure that is suitable for single-molecule trajectories.

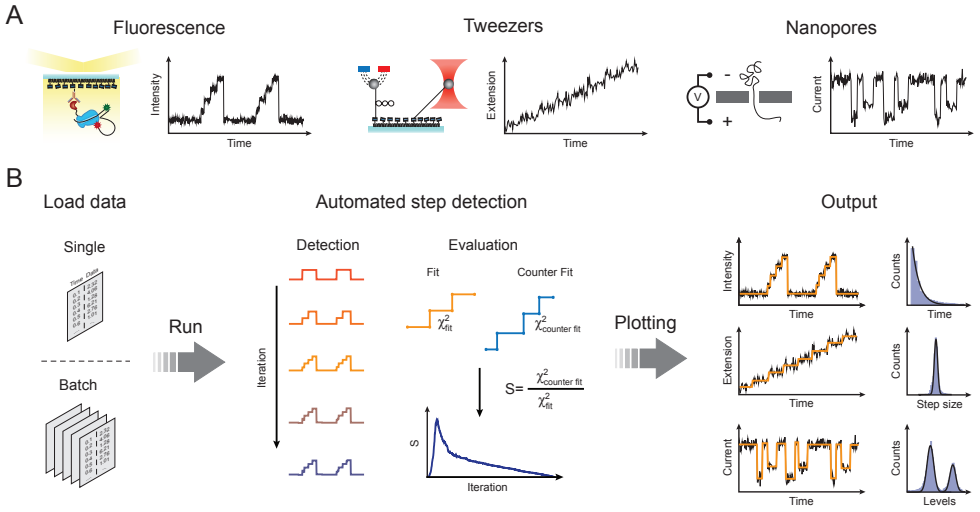

Figure 6.1: Workflow of the automated *Stepfinder*

(A) The stepfinder algorithm can be applied on a wide variety single-molecule of trajectories, including single-molecule fluorescence, magnetic & optical tweezers and nanopore data. **(B)** The algorithm requires input in the form of one or multiple .txt files with two columns (time and data). After pressing run, the algorithm iteratively adds single steps to the data that minimize χ^2. For each iteration, the quality is assessed by means of a secondary counter fit. Lastly, the best fit is selected and the algorithm outputs the corresponding fit, dwell-times, step sizes and levels. Fitting large data sets (>10^6 datapoints) can be done in less than 1 minute with a desktop computer.

6.3 Results

6.3.1 Overview of the procedure

The workflow for the *Stepfinder* algorithm is outlined in Figure 6.1. After single-molecule trajectories have been obtained, the step finding procedure can be divided in three basic steps: loading of the data, step detection and output of the result. The *Stepfinder* algorithm can be run on multiple files (batch mode) or on a single data file (Figure 6.1A). After loading the data, *Stepfinder* iteratively executes a series of partition events that allows the algorithm to determine the optimal fit for the data (Figure 6.1B). During each partition event the algorithm calculates chi-squared (χ^2), which provides a means to determine the variance between fit and the data. For each iteration, the next step is fitted at a location that yields the biggest reduction in χ^2. Subsequently, *Stepfinder* evaluates the quality of the fit by performing a secondary fit (called a counter fit) (Figure 6.1B) [20]. Once the optimal fit is determined the algorithm outputs several files that allow post-processing of the results (Figure 6.1B).

6.3.2 Step fitting

The *Stepfinder* algorithm fits data through a series of partition events that minimize chi-squared (χ^2). To fit data, the algorithm makes the sole assumption that the data contains steps with variable size (Δ) and plateau length (N) that are subject to noise (σ^2) (Figure 6.2A). The algorithm initiates the fitting procedure by splitting the data at a location that gives the lowest value of χ^2. This initial partition event generates a fit with two plateaus at a position that represents the average of the data points within the plateau (Figure 6.2A) [20]. After the first fit, the plateau that exhibits a step yielding the largest reduction χ^2 is selected for the next partition event, resulting in a fit with three plateaus (Figure 6.2A, dashed red line). The algorithm continues this process of adding a single-step to one of the plateaus for each iteration (Figure 6.2B, cyan arrow heads), until *Stepfinder* finds the user defined maximum number of steps.

Stepfinder successively selects a previously fitted step for the next partition event based on the biggest reduction in χ^2 (Figure 6.2A). By iteratively prioritizing the next fit that gives the biggest reduction in χ^2, the most prominent features of the data are fitted first followed by fits for the more refined features. As this process continues until the user defined number of steps are found, the number of step fits is likely to go beyond the 'optimal fit' (Figure 6.2B, middle). This result in 'over fitting', where new steps are fitted within the noise of the data (Figure 6.2B, bottom). To determine to optimal fit for a given dataset, it is important to evaluate the quality of the fit for every step that is added to the fit (Figure 6.1).

The quality of the existing fit is evaluated by performing a secondary fit for each iteration, hereafter called a counter fit [20]. *Stepfinder* generates counter fits by means of three steps: (i) *Stepfinder* first determines the next partition location (i_{next}) within each plateau (Figure 6.3A); (ii) next the algorithm rejects the existing step locations; (iii) *Stepfinder* builds a new fit based on the i_{next} locations, generating new plateaus with a position that represents the average of the data points within each plateau (Figure 6.3A). These three steps result in a counter fit with steps that are all

located in between the existing best-fit locations (Figure 6.3A). If the analyzed data does not display step-like behavior, both the existing fit and counter fit will have similar values of χ^2 [20]. However, when the data does display step-like behavior, counter fitting results in a fit that is much worse than the existing fit (Figure 6.3A) and thereby yields a larger value of χ^2 [20].

To evaluate the quality of the a fit, the *Stepfinder* algorithm takes advantage of the changing χ^2 landscape upon counter fitting. The quality of a fit (S), can be quantified by taking the ratio of the χ^2 from the existing fit and the counter fit, which is defined as:

$$S = \frac{\chi^2_{counter\ fit}}{\chi^2_{existing\ fit}}$$

If the existing fit is at the optimal number of iterations, the χ^2 of the existing fit approximates the noise in the data, whereas the χ^2 of the counter fit reaches its maximum value ($\sim\Delta^2/4\sigma^2$). Thereby, the maximum S-value (S^{max}) can be described by: $S^{max} = 1+P$, where P equals the maximum value of the counter fit ($\Delta^2/4\sigma^2$). The strong difference of χ^2 between the fit and the counter fit when an optimal number of iterations is reached, results in a S-value that is much larger than one (Figure 6.3). In contrast, when the data is under fitted, the χ^2 of the counter fit and existing fit approximate each other, resulting in a S-value that is close to one (Figure 6.3B).

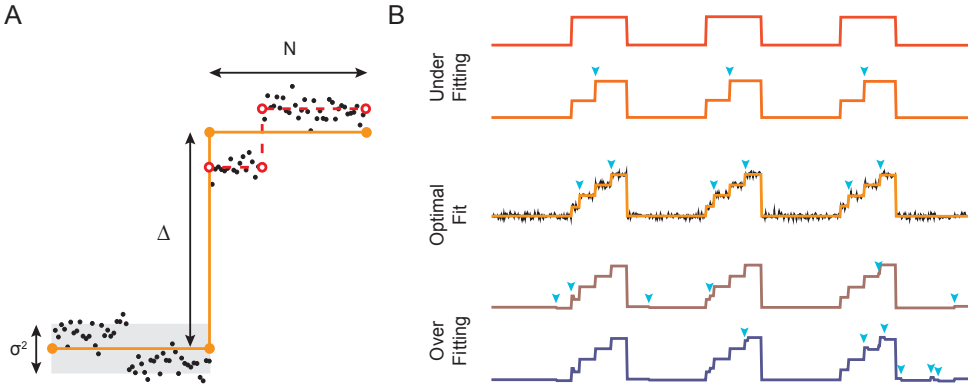

Figure 6.2: Global arrangement of the *Stepfinder* algorithm

(A) An example of an iterative step fit (orange line) on a single-molecule trajectory (black dots). Single-molecule trajectories are fitted by the *Stepfinder* algorithm by iteratively minimizing chi-squared (χ^2). To perform a step fit the program makes the assumption that the data contains steps (Δ), bounded by a plateau (N) that is subject to noise (σ^2, grey box). After the first step fit, *Stepfinder* selects the plateau with the largest value of χ^2, for the next partition event (red dotted lines). This process continues until the user defined number of steps is reached. **(B)** An example of the iterative process of step fitting by the *Stepfinder* algorithm. The algorithm successively adds a single step to the data (cyan triangles) and thereby minimizes χ^2. Step fitting below the optimal number of steps is considered under fitting, whereas step fitting beyond the optimal number of steps is considered over fitting.

Similarly, over fitting a dataset with steps that follow the noise, only results in a marginal change in the χ^2 of the counter fit (Figure 6.2B & Figure 6.3B). Therefore, the S-curve is a powerful indicator for the quality of the fit, displaying a sharp peak when the optimal fit is reached (Figure 6.3B).

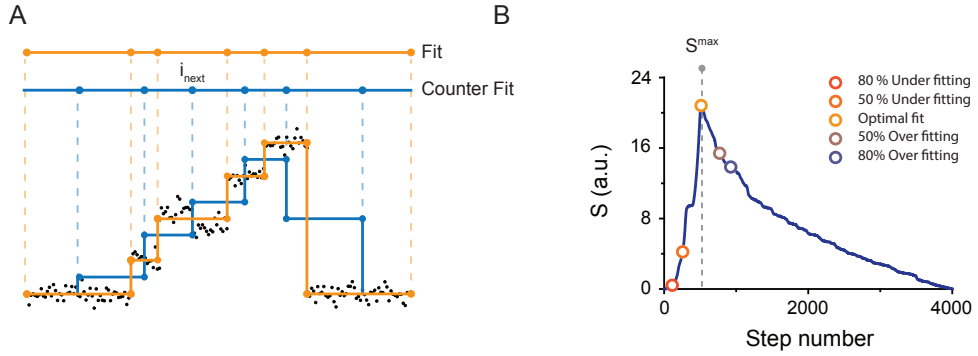

Figure 6.3: Determining the quality of a step fit

(A) For every step fit the algorithm performs, the quality of the fit (orange line) is evaluated by means of a secondary fit (blue line, called a counter fit). The counter fit is built by determining the next partition point (i_{next}), after which the current is rejected. Subsequently, the algorithm places the counter fit (blue) plateaus at a located within the existing fit (orange). **(B)** A representative example of an S-curve. The S value can be calculated by taking the χ^2 of the fit and dividing it by the χ^2 of the counter fit. When the existing fit is close to the optimal fit, the counter fit is at its worst, yielding a large value of S. However, when the data is over- or under fitted, little change in χ^2 is observed, resulting in S values close to 1. Thereby, the S-curve is a powerful predictor of the quality of the fit, exhibiting a sharp peak when the optimal number of step fits is performed.

6.3.3 A multi-pass strategy for automated step fitting

The S-curve is a robust measure to determine the quality of a fit, showing a distinct peak when the optimal number of iterations is reached. When the data exhibits steps that are in the same order of size and duration (e.g. Δ_1 or Δ_2, Figure 6.4A & Figure 6.4B), the optimal fit could be determined by finding the global maximum of the S-curve (S^{max}) (Figure 6.4D). However, this assumption cannot be made when the data exhibits steps that vary widely in size and duration (e.g. Δ_1 and Δ_2, Figure 6.4C). In this case, the S-curve exhibits a secondary peak (S_{P2}) that has a lower S_{P2}^{max} than the first peak (S_{P1}) (Figure 6.4D). Notably, the position of these peaks is identical to the peaks observed for a dataset with either Δ_1 or Δ_2 (Figure 6.4D). In this case, a significant portion of the refined steps would not be fitted, when the optimal fit could be determined by finding S^{max}.

To automate step detection, we developed a multi-pass strategy that determines the optimal fit for the data over two rounds. The *Stepfinder* algorithm first performs a step-fit, which yields a S-curve with a global maximum that corresponds to the most prominent features in the data. This step fit is then subtracted from the data and a secondary step-fit is performed on the 'residual data'. Only if the global maximum of the secondary step-fit is above the user defined threshold, coined acceptance

threshold, the fit will be accepted (Figure 6.6D). In summary, the multi-pass approach combined with the acceptance threshold on the second round of fitting provides a robust method for automated step detection.

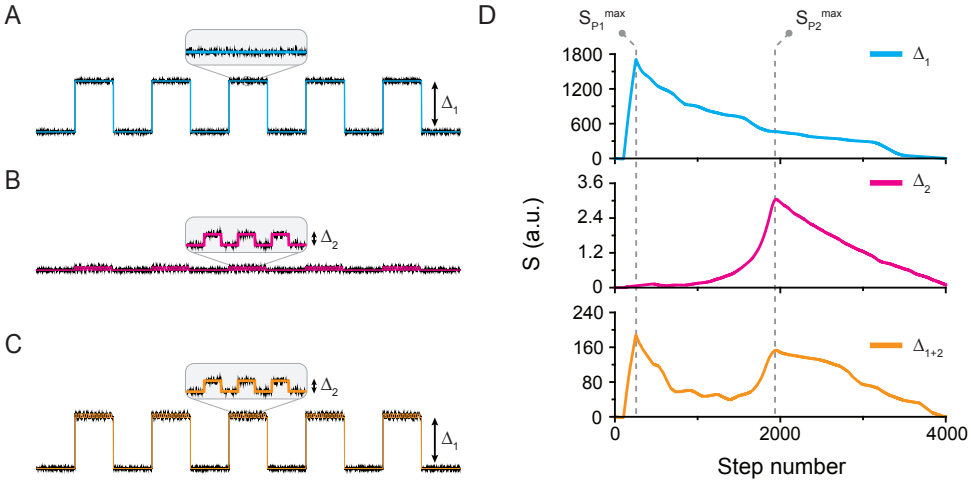

Figure 6.4: Multi-pass step detection to determine the optimal fit

(A) An example trace displaying uniform steps with a size of Δ_1. **(B)** An example trace displaying uniform steps with a size of Δ_2. **(C)** An example trace displaying non-uniform steps with a size of Δ_1 and Δ_2. **(D)** S-curves for the three example traces displayed in [A-C]. The global maximum of peak 1 (S_{P1}) and peak 2 (S_{P2}) are indicated with a dotted grey line. The S-curve for the data set with both large (Δ_1) and small (Δ_2) steps exhibits two peaks. If the small steps of Δ_2 are considered as an increased noise level of Δ_2, the global maximum of peak 1 (S_{P1}^{max}) can be described by $S_{P1}^{max}=1+P_1$, with P_1 is $\Delta_1^2/4(\sigma^2+1/4\Delta_2^2)$. This results in a S-curve with an S^{max} that is located at the same number of iterations as for a data set that exhibits only steps of Δ_1 or Δ_2, albeit lowered. However, if both the steps (Δ_1 and Δ_2) are fitted over the noise, the S-curve shows a secondary peak that can be described by: $S_{P1,2}^{max}=1+P_{1,2}$, with $P_{1,2}=\Delta_{1,2}^2/4\sigma^2$ and $\Delta_{1,2}$ is the weighted average of the large and the small step fractions.

6.3.4 An enhanced algorithm for automated step dectection

For each iteration, the *Stepfinder* algorithm selects an existing plateau (N_w) and splits it into a left (N_L) and right (N_R) plateau (Figure 6.5A). The position of these newly acquired plateaus is strongly dependent on the location of the partition point within N_w (Figure 6.5A). The average position (A) of a plateau (e.g. N_L) for any given location (i) can be described by:

$$A_L = \frac{1}{N_L}\sum_{i=1}^{N_L} x(i)$$

The iterative nature of determining this partition point, requires a substantial amount of computing power and becomes problematic when analyzing large datasets (e.g. >1·10^6 data points) (Figure 6.5B). Previously, *Stepfinder* determined the next partition point of N_w by calculating the χ^2 for all possible locations (i), selecting the step-fit that yields the

largest reduction in χ^2. However, this means that for a dataset with N_0 data points, the algorithm performs N_0^2 single x(i) operations to determine a single partition point. Next, the algorithm would repeat the same cycle to determine the partition point within the newly generated left (N_L) and right (N_R) plateau. With this scheme, it requires $2 \cdot (N_0^2/2)$ x(i) operations to locate the next two partition points. This cycle of partitioning continues plateaus until the algorithm find the user defined number of steps, which roughly scales with $2 \cdot N_0^2$ ((1+½+¼+…) $\cdot N_0^2$) operations per data set. Thereby, the required computing time significantly increases with an increase in the number of data points in a dataset (Figure 6.5B).

To reduce the (i) operations that are required to fit a dataset, we completely re-organized the code and streamlined the iteration process. A strong reduction in the number of required (i) operations can be made by re-using the information that is obtained during the localization of the first partition point. After the algorithm has determined the average (A_w) value of a plateau (N_w), the new version of *Stepfinder* determines the location of both N_L and N_R for x(i), using a single operation. The procedure starts with x(1) that is located at the left side of N_w (Figure 6.5A). The location (A_L) of N_L can be deduced by $A_L(i)=x(i)$, whereas the level of N_R is defined by:

$$A_r(\mathrm{i}) = \frac{(N_w \cdot A_w - x(i))}{(N_w - 1)}$$

This procedure is repeated for the next location (i+1) until each location of N_w is calculated, requiring only N_0 operations per plateau. For a whole data set this roughly scales with $2 \cdot N_0$, which is a gain of a factor of N_0 compared to the previous algorithm. Depending on the size of the analysed dataset, this improvement yields a speed gain of several orders of magnitude (Figure 6.5B). Notably, Additional speed can be gained by only saving a minimal pair of parameters for each step-fit operation.

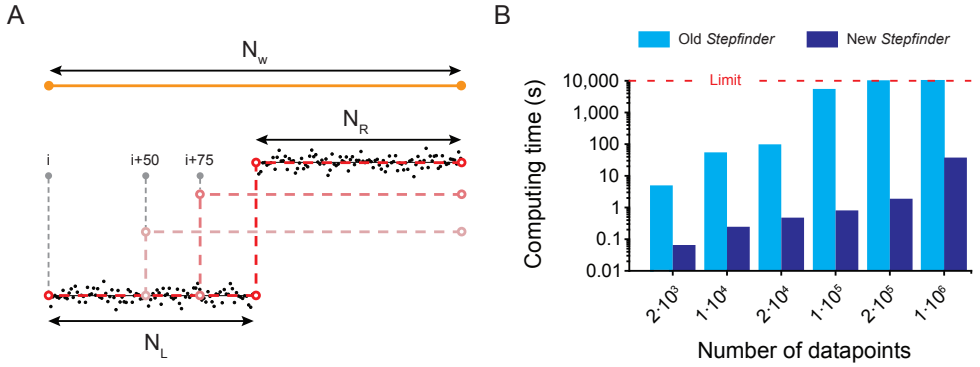

Figure 6.5: An enhanced *Stepfinder* for high-throughput step detection

(A) An example of the iterative nature of the step fit procedure. The existing plateau (N_w, orange line) is partitioned into two new plateaus (N_L and N_R, dark red dotted) at a point that yields the largest reduction in χ^2. To determine this partition point, the algorithm iteratively calculates χ^2 for each data point, starting at i until all data points of N_w have been calculated (e.g. i+50, faded red dotted lines). **(B)** Comparison of the old and new version of the *Stepfinder* algorithm. The algorithms were tested by measuring the computing time of various datasets on a desktop computer. The red dotted line indicates the limit that was set for the computing time.

6.3.5 Step fitting of experimental data

The newly developed automated multi-pass *Stepfinder* algorithm was applied on traces from the CRISPR-associated Cas3 helicase [37–40], which could not be analyzed with the previous version of *Stepfinder*. A detailed description on the experimental procedures and analysis are described in Chapter 3 on page 73. In brief, DNA bound Cas3 molecules were presented with ATP to initiate DNA unwinding. The fluorophores on the DNA substrate were able to report on DNA unwinding though an increase in FRET (Figure 6.6A). Before ATP was added, the labelling positions on the DNA yielded a FRET value that was indistinguishable from the background noise. Upon addition of ATP, a gradual increase in FRET was observed (Figure 6.6B).

The unwinding events of Cas3 were marked by plateaus (Figure 6.6B & Figure 6.6C), suggesting that Cas3 unwinds the DNA in discrete steps. Besides the increase in FRET that reports on unwinding, slipping events, in which the DNA abruptly moves backwards and reanneals (Figure 6.6C) were observed. The unwinding events using the automated multi-pass *Stepfinder* algorithm. The first round of the step fitting yielded a sharp peak in the S-curve (Figure 6.6D), whereas the second round of step fitting yielded a global maximum that was below the threshold. This indicates that the detected steps were in the same order of unity. A histogram of the FRET levels exhibited four equally spaced levels, suggesting that the helicase unwinds the DNA in discrete steps (Figure 6.6E) (Chapter 3 on page 73). This example shows that the enhanced version of the *Stepfinder* algorithm is able to automatically detect steps in baseline type trajectories without any prior knowledge on the number of states in the data.

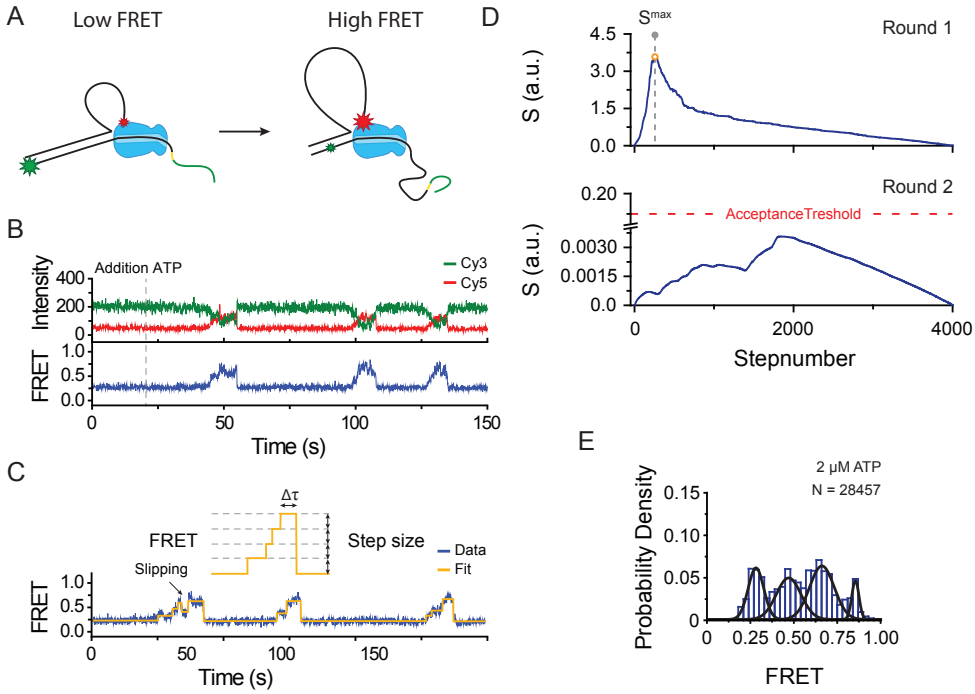

Figure 6.6: Application of the enhanced *Stepfinder* on experimental data

(A) Schematic of loop formation by the CRISPR-associated Cas3 helicase/nuclease protein (blue). The appearance of FRET during loop formation is indicated by the size of the star: low FRET, large green star) or high FRET, large red star. **(B)** A representative time trace of donor (Cy3, green) and acceptor (Cy5, red) fluorescence and corresponding FRET (blue) exhibiting multiple unwinding events. ATP (2 μM) was added at t = 20s (dashed gray line). **(C)** Representative FRET trace (dark blue) fitted with the enhanced *Stepfinder* algorithm (orange). **(D)** S-curve for two rounds of fitting on the dataset with unwinding events of Cas3. The global maximum of the S-curve for round two (bottom) was below the set acceptance threshold and therefore the second round of fitting was not executed. **(E)** Distribution of FRET levels obtained through the step-finder algorithm. Black lines represent a Gaussian fit.

6.4 References

1 C. Joo, M. Fareh, V. Narry Kim, Bringing single-molecule spectroscopy to macromolecular protein complexes. *Trends Biochem. Sci.* **38**, 30–37 (2013).

2 R. A. Forties, M. D. Wang, Minireview Discovering the Power of Single Molecules. *Cell.* **157**, 4–7 (2014).

3 M. F. Juette *et al.*, The bright future of single-molecule fluorescence imaging. *Curr. Opin. Chem. Biol.* **20**, 103–111 (2014).

4 K. C. Neuman, A. Nagy, Single-molecule force spectroscopy : optical tweezers , magnetic tweezers and atomic force microscopy. *Nat. Methods.* **5**, 491–505 (2008).

5 T. Ha, Single-molecule methods leap ahead. *Nat. Methods.* **11**, 1015–1018 (2014).

6 V. Aggarwal, T. Ha, Single-molecule fluorescence microscopy of native macromolecular complexes. *Curr. Opin. Struct. Biol.* **41**, 225–232 (2016).

7 M. Fareh *et al.*, Single-molecule pull-down for investigating protein-nucleic acid interactions. *Methods* (2015).

8 T. R. Blosser *et al.*, Two distinct DNA binding modes guide dual roles of a CRISPR-cas protein complex. *Mol. Cell.* **58**, 60–70 (2015).

9 S. D. Chandradoss, N. T. Schirle, M. Szczepaniak, I. J. Macrae, C. Joo, A Dynamic Search Process Underlies MicroRNA Targeting. *Cell.* **162**, 96–107 (2015).

10 K. Yeom *et al.*, Single-molecule approach to immunoprecipitated protein complexes: insights intomiRNA uridylation. *EMBO Rep.* **12**, 690–696 (2011).

11 I. F. Gallardo *et al.*, High-Throughput Universal DNA Curtain Arrays for Single-Molecule Fluorescence Imaging. *Langmuir.* **31**, 10310–10317 (2015).

12 B. Sun, M. D. Wang, Single-molecule perspectives on helicase mechanisms and functions. *Crit. Rev. Biochem. Mol. Biol.* **9238**, 1–11 (2015).

13 T. Ha, A. G. Kozlov, T. M. Lohman, Single-molecule views of protein movement on single-stranded DNA. *Annu. Rev. Biophys.* **41**, 295–319 (2012).

14 R. Vlijm *et al.*, Nucleosome Assembly Dynamics Involve Spontaneous Fluctuations in the Handedness of Article Nucleosome Assembly Dynamics Involve Spontaneous Fluctuations in the Handedness of Tetrasomes. *Cell Rep.* **10**, 216–225 (2015).

15 S. J. Heerema, C. Dekker, Graphene nanodevices for DNA sequencing. *Nat. Nanotechnol.* **11**, 127–136 (2016).

16 C. Dekker, Solid-state nanopores. *Nat. Nanotechnol.* **2**, 209–215 (2007).

17 L. Ma, S. L. Cockroft, Biological Nanopores for Single-Molecule Biophysics. *ChemBioChem.* **11**, 25–34 (2010).

18 C. Plesa *et al.*, Fast Translocation of Proteins through Solid State Nanopores. *Nano Lett.* **13**, 11–16 (2013).

19 S. A. Mckinney, C. Joo, T. Ha, Analysis of Single-Molecule FRET Trajectories Using Hidden Markov Modeling. *Biophys. J.* **91**, 1941–1951 (2006).

20 J. W. J. Kerssemakers *et al.*, Assembly dynamics of microtubules at molecular resolution. *Nature.* **442**, 709–712 (2006).

21 S. A. Mckinney, D. M. J. Lilley, T. Ha, Structural dynamics of individual Holliday junctions. *Nat. Struct. Mol. Biol.* **10**, 93–97 (2003).

22 I. Sgouralis, S. Presse, Biophysical Perspective An Introduction to Infinite HMMs for Single-Molecule Data Analysis. *Biophys. J.*, 2021–2029 (2017).

23 S. C. Kuo, J. Gelles, E. Steuer, M. P. Sheetz, A model for kinesin movement from nanometer-level movements of kinesin and cytoplasmic dynein and force measurements. *J. Cell Sci.* **14**, 135–138 (1991).

24 K. Svoboda, C. F. Schmidt, B. J. Schnapp, S. M. Block, Direct observation of kinesin stepping by optical trapping interferometry. *Nature.* **365** (1993).

25 M. Blanco, N. G. Walter, *Analysis of Complex Single-Molecule FRET Time Trajectories* (Elsevier Inc., ed. 1, 2010; http://dx.doi.org/10.1016/S0076-6879(10)72011-5), vol. 472.

26 T. S. Ferguson, A bayesian analysis of some nonparametric problems. *Ann. Stat.* **1**, 209–230 (1973).

27 K. E. Hines, A primer on bayesian inference for biophysical systems. *Biophys. J.* **108**, 2103–2113 (2015).

28 K. E. Hines, J. R. Bankston, R. W. Aldrich, Analyzing single-molecule time series via nonparametric bayesian inference. *Biophys. J.* **108**, 540–556 (2015).

29 B. C. Carter, M. Vershinin, S. P. Gross, A comparison of step-detection methods: how well can you do? *Biophys. J.* **94**, 306–319 (2008).

30 S. Myong, M. M. Bruno, A. M. Pyle, T. Ha, Spring-Loaded Mechanism of DNA Unwinding by Hepatitis C Virus NS3 Helicase. *Science.*, 513–517 (2007).

31 R. T. Dame, M. C. Noom, G. J. L. Wuite, Bacterial chromatin organization by H-NS protein unravelled using dual DNA manipulation. *Nature.* **444**, 387–390 (2006).

32 S. L. Reck-Peterson *et al.*, Single-Molecule Analysis of Dynein Processivity and Stepping Behavior. *Cell.* **126**, 335–348 (2006).

33 M. A. Beuwer, M. W. J. Prins, P. Zijlstra, Stochastic protein interactions monitored by hundreds of single-molecule plasmonic biosensors. *Nano Lett.* **15**, 3507–3511 (2015).

34 R. Vlijm, J. S. J. Smitshuijzen, A. Lusser, C. Dekker, NAP1-Assisted Nucleosome Assembly on DNA Measured in Real Time by Single-Molecule Magnetic Tweezers. *PLoS One.* **7**, 1–11 (2012).

35 B. T. Harada *et al.*, Stepwise nucleosome translocation by RSC remodeling complexes. *Elife.* **5**, 1–20 (2016).

36 H. Isojima, R. Iino, Y. Niitani, H. Noji, M. Tomishige, Direct observation of intermediate states during the stepping motion of kinesin-1. *Nat. Chem. Biol.* **12**, 290–297 (2016).

37 T. Sinkunas *et al.*, Cas3 is a single-stranded DNA nuclease and ATP-dependent helicase in the CRISPR/Cas immune system. *EMBO J.* **30**, 1335–1342 (2011).

38 S. Mulepati, S. Bailey, In vitro reconstitution of an Escherichia coli RNA-guided immune system reveals unidirectional, ATP-dependent degradation of DNA Target. *J. Biol. Chem.* **288**, 22184–22192 (2013).

39 Y. Huo *et al.*, Structures of CRISPR Cas3 offer mechanistic insights into Cascade-activated DNA unwinding and degradation. *Nat. Struct. Mol. Biol.* **21**, 771–7 (2014).

40 R. N. Jackson, M. Lavin, J. Carter, B. Wiedenheft, Fitting CRISPR-associated Cas3 into the Helicase Family Tree. *Curr. Opin. Struct. Biol.* **24**, 106–114 (2014).