User manual

General notes on experimental data for optimal step fitting

AutoStepfinder is capable of detecting steps in trajectories of various techniques, including single-molecule fluorescence, nanopores, and magnetic and optical tweezers. While the details of these experimental approaches differ, we provide a set of general guidelines that will maximize the performance of *AutoStepfinder*.

A. **Sampling rate:** *AutoStepfinder* determines the significance of steps based on the number of data point in the plateau (N_i) and the size of the step (Δ) (Figure 3). Thereby, the sampling rate which the single-molecule measurement is performed, i.e. the number of data points per time-unit, is an important factor in step fitting. To facilitate step fitting by *AutoStepfinder*, it is recommended to maximize the number of independent data points per plateau by acquiring data at high sampling rates. ▲Critical Increasing the sampling rate in single-molecule measurements may come at a cost. For example, an increased sampling rate in single-molecule fluorescence measurements will require higher laser powers to collect a large number of photons per frame. These high laser powers will induce fast photobleaching and thereby limit the observation time of the experiment. In addition, a similar upper sampling limit exists for response time-limited systems, such as in magnetic/ optical tweezer and nanopore experiments.

B. **Drift:** A commonly found artefact in single-molecule trajectories is drift or other types of movement in the x-, y- or z-direction. These movements result in trajectories with a gradually decreasing or oscillating signal, which may interfere with the performance of *AutoStepfinder*. Therefore, it is recommended to limit drift during the measurements as much as possible and discard traces from the analysis that show an excessive amount of drift.

C. **Filtering of data:** In single-molecule data analysis, it is a common practice to reduce the noise in the trajectories by smoothing the data with moving averages and filters. However, the use of these filters may also smooth the state-to-state transitions. Given that *AutoStepfinder* works best on instant state-to-state transitions, care should be taken when applying such filters. Indeed, it is advised to run *AutoStepfinder* on raw data or otherwise consider using a step preserving filter, such as a median or Chung-Kennedy filter.

Initializing the *AutoStepfinder* algorithm and auxiliary tools ●Timing: 1-10 min

1. Start MATLAB as described by Mathworks.
2. Copy all the files enclosed in *AutoStepfinder* folder to the working directory of MATLAB. Alternatively, change the initial working directory of MATLAB by going Home tab then Preferences > General > "Initial working folder" and specify the full path to the *AutoStepfinder* folder.

Formatting data for step detection by *AutoStepfinder* ●Timing: 5-10 min

3. *AutoStepfinder* runs on a single-column text file (.txt) that encompasses numeric single-molecule data. Alternatively, *AutoStepfinder* can run two-column text files (.txt) with the time axis in the first column and data in the second column. In the latter case, the time axis will be ignored during the step-fitting procedure.
4. To ensure *AutoStepfinder* runs properly, the input files for *AutoStepfinder* should be free of non-numeric values, including: ‘infinity values’ (Inf) and ‘not a number’ (NaN). To remove these values one can use the *DataDuster* auxiliary tool, which is located in the *AutoStepfinder* package.
5. Start the *DataDuster* auxiliary tool by opening DataDuster.m and running the code in the editor tab or the command window of MATLAB.
6. After pressing “Run”, a graphical user interface (GUI) will appear, in which the user can adjust the run settings for *DataDuster* (Figure 7).

   For troubleshooting, see table 2

   6A. *Data Path*: The Data Path box allows one to specify the directory of the input data. ▲Critical By default, the directory is set to the current directory of MATLAB. The default location of Data Path can be changed at line 47 of the DataDuster.m file.

   6B. *Run Mode*: Run mode specifies whether one runs *DataDuster* on a single file or run all files in a selected folder. To run a single .txt file, check single and run the algorithm to select a file. For batch style processing, check batch and *DataDuster* will analyze all .txt files in the specified directory.

   6C. *Columns to clean and save*: The columns to clean and save box allows one to specify on which column to run *DataDuster*. By default, it runs through all columns in the loaded .txt file(s). To specify a specific column, uncheck the “all columns” box and specify the number of the column to analyze.

6D. *Replace non-numeric values with*: The replace non-numeric values with box allows one to specify what to do with the non-numeric values.

6.D1. *Neighbor*: Replaces non-numeric values with the mean value of the neighboring two data points.

6.D2. *Mean*: Replaces non-numeric values with the mean value of the dataset.

6.D3. *Median*: Replaces non-numeric values with the median value of the dataset.

6.D4. *Remove*: Removes non-numeric values from the dataset.

6E. *Clean data:* The clean data button initiates *DataDuster* data cleaning procedure.

7. To start the data cleaning procedure, press the "Clean data" button, located on the bottom of the GUI (Figure 7).
8. Browse to the directory of interest and select the file (single run) or the folder (batch run) that encompasses the single-molecule data for data cleaning.
9. The output of the *DataDuster* is saved in a new folder called "cleaned_data_method", where method refers to the input of the "replace non-numeric values with" box. This folder is generated in the directory of the input file. If a multi-column .txt file was loaded, *DataDuster* will output each column as a separate .txt file named: filename_col_0x.txt, which can be directly loaded into the *AutoStepfinder* algorithm. Notably, the MATLAB console will display the number of replaced values in each column. ▲Critical *DataDuster* does not export trajectories that exhibit equidistant increase in signal, e.g. the time axis of trajectories or indices, as these trajectories are featureless and will not result in step detection by the *AutoStepfinder* algorithm.

Startup and Graphical user interface of *AutoStepfinder* ●**Timing:** **1 min**

10. Start *AutoStepfinder* by opening AutoStepfinder.m and running the code in the editor tab or the command window of MATLAB.
11. After pressing "Run", a GUI will appear, in which the fitting procedure will be executed and fitting parameters can be adjusted (Figure 8).

11A. *Fitting Window*: The top half of the GUI comprises a fitting window. This window allows one to visually inspect the fit after executing the *AutoStepfinder* algorithm. The data is displayed in blue and the corresponding fit in orange.

11B. *Data Path*: The Data Path box allows one to specify the directory of the input data. ▲Critical By default, the directory is set to the current directory of MATLAB. The default location of Data Path can be changed at line 38 of the AutoStepfinder.m file.

11C. *Run Mode*: Run mode specifies whether one runs *AutoStepfinder* on a single file or run all files in a selected folder. To run a single .txt file, check single and run the algorithm to select a file. For batch style processing, check batch and *AutoStepfinder* will analyze all .txt files in the specified directory.

11D. *Run Settings*: The Run Settings box provides a minimal set fitting of parameters that allows one to tune the fitting procedure (Figure 8).

11.D1. *Sensitivity*: The sensitivity parameter determines to what extent the algorithm fits beyond or below the optimal fit (Figure 3B). By default, the sensitivity is set to a value 1, at which the algorithm determines the optimal fit based on the S_{max} of the S-curve (Figure 4B). Typically, the sensitivity parameter ranges between 0.5 - 2.0. ▲Critical If the user has other criteria for determining the optimal number of steps, the sensitivity parameter can be adjusted. For example, the fit can be 10% over- or under fitted by increasing or decreasing the accuracy by 0.1, respectively (Figure 4B). For other adjustment possibilities see advanced options (Step 11.E).

11.D2. *Iteration range*: The iteration range parameter determines to what extent *AutoStepfinder* continues the fitting procedure. Once the number defined by the iteration range is found, the algorithm stops partitioning plateaus to minimize χ_2 and determines the optimal fit. For datasets with limited step numbers, the iteration range parameter can be decreased to reduce the computing time of the fitting procedure. Typically, the initial iteration range is set to ¼ of the number of data points of the input data.

11.D3. *Time resolution*: The time resolution parameter corresponds to the temporal resolution (e.g. the time interval between each data point) of the measurement. This parameter will be used for the time data in the output files of

AutoStepfinder. ▲Critical If a file with two columns is provided, time and data, the time column (first column) is ignored by the *AutoStepfinder* algorithm.

11E. *Adv(anced) Options*: Enabling the advanced option box displays the advanced setting of *AutoStepfinder* (Figure 8). ▲Critical It is noted that these settings are intended for advanced users that have full understanding on the step fitting procedure.

11.E1. *Accept(ance) thresh(old)*: The acceptance threshold sets a threshold for each fitting round and is compared to S_{max}-1. If the S-curve of the first or second fitting round provides a S_{max} that lays below the threshold, this fitting round will not be executed. Typically, the acceptance threshold ranges between: 0.1 – 1. ▲Critical Care should be taken when adjusting the acceptance threshold. If the acceptance threshold is set above the S_{max} of the first round of fitting, *AutoStepfinder* will not execute the step-finding procedure.

11.E2. *Manual mode*: Manual mode allows one to define the number of steps that have to be found by the algorithm. Typically, the number of manually fitted steps does not exceed the iteration range. ▲Critical Notably, when manual mode is engaged the sensitivity parameter is blocked.

11.E3. *S-Curves*: When S-Curves are turned on, the *AutoStepfinder* will display the S-curves of the first and second round in a separate window.

11.E4. *Fitting*: The fitting box allows one to choose how the position of the plateaus of the final fit is determined. By default, *AutoStepfinder* uses the averages of each plateau to determine its position for each iteration and the final fit. However, in some cases (e.g. when data exhibits spikes), one may choose to build the final fit using the median of each plateau, by checking the median parameter.

11.E5. *File ext(ension)*: The file extension box allows one to select the file type *AutoStepfinder* outputs. When .txt is checked, *AutoStepfinder* outputs text files. If .mat is checked, *AutoStepfinder* outputs matlab files, which can be used to further post-process the output of *AutoStepfinder* in Matlab.

11.E6. *Output files:* The output files box allows one to select which output files *AutoStepfinder* saves. By saving a subset of output parameters, additional speed can be gained, which may be preferred for large datasets or when optimizing fitting settings.

11F. *User plot*: The user plot box allows one to quickly assess the fitting result of the *AutoStepfinder* algorithm. By turning the *User plot* function on, *AutoStepfinder* will plot the step size, step levels and dwell-time histograms in a separate window.

11G. *Post proc(essing)*: The post processing box allows one to discard the baseline plateaus from the fit. All fitted plateaus that have a value below the provided threshold in units of the input data are removed from the "filename_properties" output file.

11H. *Run*: The run button initiates *AutoStepfinder* step fitting procedure.

Running *AutoStepfinder* ●Timing: 1-10 min

12. To start the step-finding procedure press the "Run" button, located on the right side of the GUI (Figure 8).
13. Browse to the directory of interest and select the file (single run) or the folder (batch run) that encompasses the single-molecule data for *AutoStepfinder*.
14. Press open to start the step-finding procedure. The progress of the *AutoStepfinder* analysis is displayed in the console of MATLAB. Once the console indicates "done!" the fitting procedure has been completed and output files have been saved.
15. The output of the *AutoStepfinder* analysis is saved in a new folder (StepFit_Result), which is generated in the directory that is provided in the datapath box of the GUI. By default, the output of *AutoStepfinder* consists of four files: "filename_fits", "filename_properties" and "filename_s_curve" and "filename_config". The "filename_fits" file consists of three columns (Table 1) and can be used to plot the data with corresponding fit. The "filename_properties" file consists of 8 columns (Table 1) and encompasses the information required to generate histograms of the step size, step levels and dwell-times. The "filename_s_curve" file encompasses the information required to plot the S-curves of the first and second round (Table 1). Lastly, *AutoStepfinder* generates a "filename_config" file that encompasses all the parameters that were used to generate the fit (Table 1). Notably, when batch mode is selected,

AutoStepfinder generates additional .JPEG files of the fit window, user plots and S-curves (Table 1).

Fine tuning the fit parameters for optimal results ●Timing: 1-10 min

16. ▲Critical *AutoStepfinder* is a robust approach for automated step detection that determines the optimal fit based on statistical arguments. However, despite the automated detection of steps, it is advised to always carefully inspect the quality of the fitting result before proceeding with post-processing of the data. The quality of the fit can be assessed by using the fitting window and the built-in controls of the GUI (e.g. zoom in/out and pan) (Figure 8, Step 11.A). Below we provide guidelines on how to interpret and fine-tune fitting parameters to obtain optimal fitting results. As a rule of thumb, it is recommended to maintain a conservative attitude towards step fitting in which it is better to miss small events rather than to introduce spurious steps by overfitting.

16A. *Underfitted data:* Data is considered underfitted when the number of detected steps by *AutoStepfinder* is significantly lower than the number of steps that are present in the data. Therefore, a hallmark for underfitted data is a fit in which a significant number of steps are missed. At the location where steps are obviously missed, the plateau of the fit deviates from the data (Figure 9A) and thereby these plateaus are generally associated with in large step errors (properties output file, column 8). Underfitting of data is typically associated with irregular features in the S-curve; therefore, it is recommended to inspect the corresponding S-curve (Advanced options, Step 11.E3). While the S-curve normally shows a sharp peak at the optimal step number, for some datasets the S-curve may have a non-canonical shape. For example, it might have a secondary peak or shoulder that represents a more realistic step number to fit.

Typically, underfitting can be prevented by adjusting the parameterization of the *AutoStepfinder* algorithm. Underfitting may occur when the final number of steps in the data is too close to the user provided iteration range (Step 11.D2) or when the S_{max} of the second round of fitting lies below the acceptance threshold (see advanced options, Step 11.E1). Thereby, underfitting can be prevented by increasing the iteration range or by lowering the acceptance threshold. Alternatively, one can determine the position of a specific feature in the S-curve (e.g. a shoulder or secondary peak) as follows:

16.A1. Select the data cursor tool from the build in controls of the S-curve plotting window (Figure D)

16.A2. Use the data cursor tool to determine the step number (X value) at which the shoulder or secondary peak in the S-curve occurs (Figure D).

16.A3. Enable to advanced settings and engage manual mode by checking on (Step 11.E2).

16.A4. Insert step number that was determined with the data cursor tool in the manual mode box (Figure D).

16.A5. Run *AutoStepfinder* with manual mode engaged.

! Caution: By engaging manual mode *AutoStepfinder* fits the user-defined number of steps to the data, bypassing the quality assessment of the *AutoStepfinder* algorithm. Therefore, the use of manual mode should always be guided by specific features of the S-curve. It is strongly discouraged to use manual mode without a compelling rationale.

16B. *Overfitted data:* Data is considered overfitted when the number of detected steps by *AutoStepfinder* is significantly higher than the number of steps that are present in the data. Therefore, a hallmark for overfitted data is a fit in which plateaus are fitted with a significant number of small steps that follow the noise of the data (Figure 9B). By fitting the noise of the data, plateaus are divided into smaller ones, which can detrimental for the outcome of the step analysis (e.g. dwell-times can be significantly shorter when data is overfitted). In most experimental contexts, it is better to miss small events than to introduce spurious small steps by overfitting.

Overfitting of data is typically associated with wrong parameterization of the *AutoStepfinder* algorithm. Overfitting of data by *AutoStepfinder* typically occurs when the user-defined acceptance threshold is set too low. As a result of the low acceptance threshold, *AutoStepfinder* will consider noise as small steps and overfit the data (Advanced options, Step 11.E1). In some cases, overfitting may occur when the user provides an iteration range that is approximately more than an order of magnitude larger than the number of steps in the data, which can be prevented by lowering the iteration range (Step 11.D2).

16C. *Correctly fitted data:* A fit describes the data well when the majority of the plateaus are fitted, while noise and other artefacts in the data are not included in the fit (Figure 9C). If one is satisfied with the fitting results proceed to step 17 of this protocol.

Post-processing of *AutoStepfinder* output ●Timing: 10-30 min

The output of *AutoStepfinder* can be post-processed to generate informative plots using any kind of spreadsheet or graphing software (e.g. OriginPro, Prism, SigmaPlot, MATLAB, Python and Excel). Below we provide a description on how the data can be processed using OriginPro.

Step-size, level and dwell-time histograms

17. Open OriginPro and load the filename_properties.txt file by going to File > Import and select Single ASCII (Figure 10A).
18. Select the filename_properties.txt in the StepFit_Result folder and click "Open".
19. Select a column of interest (e.g. column 5, StepSize) by clicking on the column header (E(Y)). The column should now be highlighted in black (Figure 10A).
20. To generate a histogram, go to Plot > Statistics and select "Histogram" (Figure 10A).
21. Double clicking on the bars of the histogram will open the Plot Details window (Figure 10B) that allows one to tune the bin size. Alternatively, right click on the bars of the histograms and select "Plot Details".
22. Uncheck "Automatic Binning", and define a bin size or a number of bins by selecting "Bin Size" or "Number of Bins", respectively (Figure 10B). As a rule of thumb, one can estimate the appropriate number of bins for a dataset by taking the square root of the number of data points in the dataset (round off if necessary).
23. Once the appropriate number of bins has been determined press "Apply" and "OK" (Figure 10B). This will generate a histogram of the selected column, for example with Levels (Figure 11B), Step size (Figure 11C) or Dwell-times (Figure 11D). Notably, for step size histograms a peak at a negative step size indicates a step from a higher level to a lower level, whereas a peak at a positive step size indicates a step from a lower to a higher level (Figure 11C).
24. To fit the histograms, the histograms need to be converted to a bar plot. To convert the histogram to a bar plot, right click on the histogram and select "Go to bin worksheet".
25. Select the "Bin Centers (X)" and "Counts (Y)" columns.
26. With the columns selected go to Plot > Column/ Bar/ Pie and select "Column" (Figure 10A).
27. This bar plot can be fitted with different functions, depending on the distribution of the data. For example, normally distributed data can be fitted with a Gauss function by going to Analysis > Peaks and Baseline > Multiple Peak Fit and selecting: "Open

Dialog”, whereas data that follows an exponential decay can be fitted by going to Analysis > Fitting > Exponential Fit and selecting: “Open Dialog” (Figure 10A).

Transition density plots

28. Open OriginPro and load the filename_properties.txt file by going to File > Import and select Single ASCII (Figure 10A).
29. Select the level before (C(Y)) column.
30. Right click on the selected column and click on: Set As > X, the column header should change from C(Y) to C(X2) (Figure 10A).
31. Select the level before (C(X2)) and level after (D(Y2)) column by clicking on the column header. The column should now be highlighted in black.
32. Bin the data in 2D by going to Statistics > Descriptive Statistics > 2D Frequency Count/ Binning and select “Open Dialog” (Figure 10A).
33. Adjust “Specify Binning Range by” to “Bin Centers” (Figure 10C).
34. Uncheck “Automatic Binning”, and define a bin size or a number of bins by selecting “Bin Size” or “Number of Bins” (Figure 10C). As a rule of thumb, one can estimate the appropriate number of bins for a dataset by taking the square root of the number of data points in the dataset, round off if necessary.
35. Repeat step 33-34 for the Y data, selecting the same parameters, such as bin size/ bin numbers (Figure 10C).
36. Press “OK” to generate a new workbook with 2D binned data (Figure 10C).
37. Select all columns of the newly generated workbook with 2D binned data.
38. With the columns selected go to Plot > Contour and select “Color Fill” (Figure 10A).
39. In the pop-up window select “Y across columns” for “Data Format” (Figure 10D).
40. Change “Y Values in” to “Column Label” (Figure 10D).
41. Make sure that in the “Bin Centers”, “LevelAfter” is selected under “Column Label” (Figure 10D).
42. Select “1st column in selection” for “X values in” (Figure 10D).
43. Press “OK” (Figure 10D) and the transition density plot will be generated (Figure 11E).
44. The contour plot can be formatted by right clicking on the center of the graph window and going to “Plot Details”.