Biophysical *Journal*
Computational Tool

Biophysical Society

POTATO: Automated pipeline for batch analysis of optical tweezers data

Stefan Buck,[1] Lukas Pekarek,[1] and Neva Caliskan[1,2,*]
[1]Helmholtz Institute for RNA-based Infection Research (HIRI), Würzburg, Germany and [2]Medical Faculty, Julius-Maximilians University Würzburg, Würzburg, Germany

ABSTRACT Optical tweezers are a single-molecule technique that allows probing of intra- and intermolecular interactions that govern complex biological processes involving molecular motors, protein-nucleic acid interactions, and protein/RNA folding. Recent developments in instrumentation eased and accelerated optical tweezers data acquisition, but analysis of the data remains challenging. Here, to enable high-throughput data analysis, we developed an automated python-based analysis pipeline called POTATO (practical optical tweezers analysis tool). POTATO automatically processes the high-frequency raw data generated by force-ramp experiments and identifies (un)folding events using predefined parameters. After segmentation of the force-distance trajectories at the identified (un)folding events, sections of the curve can be fitted independently to a worm-like chain and freely jointed chain models, and the work applied on the molecule can be calculated by numerical integration. Furthermore, the tool allows plotting of constant force data and fitting of the Gaussian distance distribution over time. All these features are wrapped in a user-friendly graphical interface, which allows researchers without programming knowledge to perform sophisticated data analysis.

SIGNIFICANCE Studying (un)folding of biopolymer structures with optical tweezers under different conditions generates very large data sets for statistical data analysis. Recent technical improvements accelerated data acquisition by coupling modern instruments with microfluidic systems, at the same time creating the need for a high-throughput and unbiased data analysis. We developed practical optical tweezers analysis tool (POTATO), an open-source python-based tool that can process data gathered by any optical tweezers force-ramp experiment in an automated fashion. POTATO is principally designed for data preprocessing, identification of (un)folding events, and the fitting of force-distance curves. In addition, all parameters for preprocessing, statistical analysis, and fitting of the curves can be adapted to suit the data set under analysis in an easy-to-use graphical user interface.

INTRODUCTION

Arthur Ashkin received the Nobel Prize in 2018 for his research on trapping dielectric particles with laser light in optical tweezers (OTs) (1). OTs enable probing of structural dynamics of individual molecules by monitoring internal forces and short-lived intermediate states in real time (2–5). This technique has been widely used to study structures of nucleic acids and dynamics of RNA/protein folding (6–10). In addition, OTs can also be used to probe the molecular interactions between small molecules, proteins, and nucleic acids (11–13). Recently, the combination of OTs with confocal microscopy enabled simultaneous measurements of force and fluorescence that provided unprecedented insights into molecular mechanisms such as timing and order of events during transcription or translation (12,14–16). Basically, in a typical OT experiment, a biopolymer, such as a protein, DNA, or RNA molecule, is tethered between two dielectric beads via labeled handles. The beads are then trapped by focused laser beams, so-called optical traps. Following this, several modes of operation are possible. In force-ramp mode, the beads are precisely displaced in a monotonous manner, which applies increasing forces onto the biopolymer (Fig. 1 *A*). Since trapped beads behave as if they were attached to mechanical springs, the applied force can be calculated from the measured displacement of the beads out of the trap focus according to Hooke's law (Fig. 1 *B*) (17). This mode is commonly used to determine the elastic properties of the molecule and/or to

Submitted November 23, 2021, and accepted for publication June 27, 2022.
*Correspondence: neva.caliskan@helmholtz-hiri.de
Stefan Buck and Lukas Pekarek contributed equally to this work.
Editor: Gijs Wuite.
https://doi.org/10.1016/j.bpj.2022.06.030

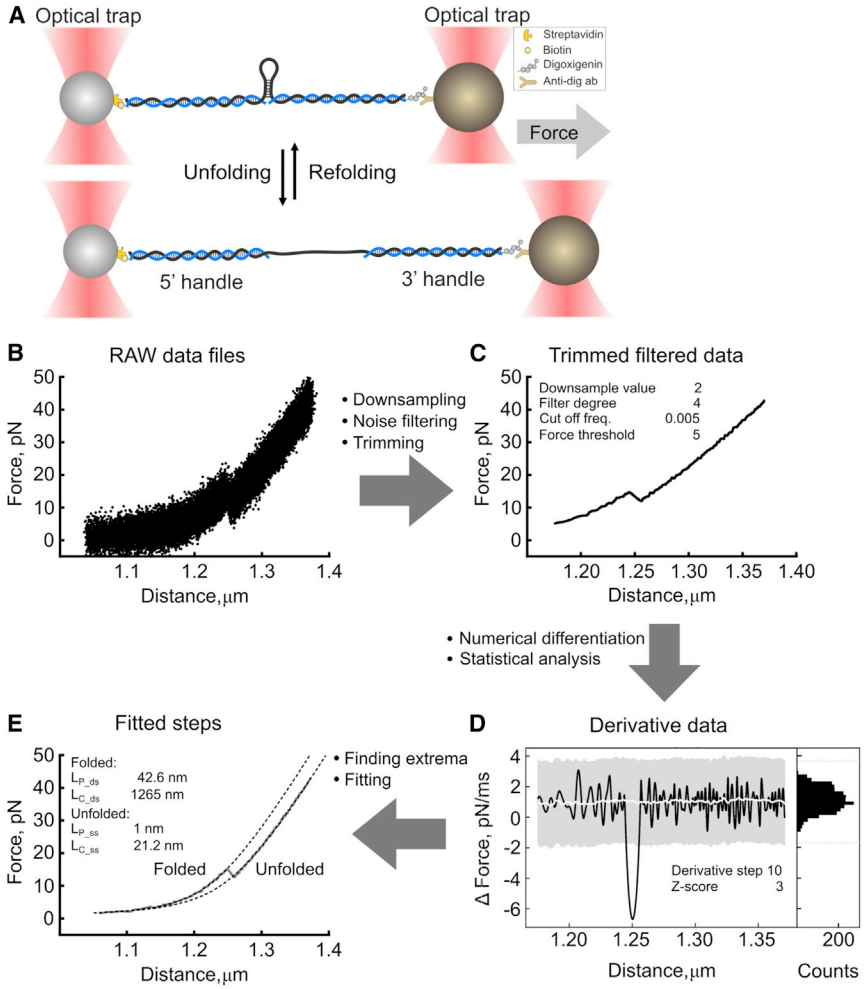

FIGURE 1 Schematic of the pipeline. (*A*) Diagram illustrating the optical tweezers experiments. RNA is hybridized to single-stranded DNA handles and immobilized on beads. These are used to exert a pulling force on the RNA with a focused laser beam. In force-ramp operation mode, the force is gradually increased until the structure in the middle is unfolded (*bottom*). Release of the force allows the structure to refold (*top*). (*B* and *C*) RAW data files (*B*) are downsampled, the noise is filtered using a Butterworth signal filter, and the data are trimmed at a minimum force threshold to yield the trimmed filtered data (*C*). (*D*) Then, the time derivative is calculated numerically to yield the derivative data; a histogram of the derivative value distribution (*right*) shows two populations—normal-like distribution represents the experimental noise, while the other population of outliers represents the (un)folding steps. The derivative data are then statistically analyzed—the standard deviation and moving median are calculated. Peaks in derivative data that exceed median (*white line*) $\pm$ *Z* score (*gray region*) are classified as (un)folding events. The beginning and end of each event are derived. (*E*) The coordinates of the events are then used to define the region for fitting, yielding the fitted steps. Finally, the output data files are exported according to the selected settings. The FD curve shown here was simulated (see supporting material). To see this figure in color, go online.

determine the rupture forces at which transitions in folding and unfolding occur.

On the other hand, a constant-force operation mode allows tracking the molecule of interest in real time as it transitions between different conformational states, yielding kinetic parameters of folding-unfolding of molecules or progressive movements of molecular motors (5). Accordingly, OT experiments also allow precise calculation of the work done on the system of interest (18,19). Previously, OT instruments were self-built by researchers, and thus application required substantial physics and engineering background. Furthermore, such experiments were highly time demanding and labor intensive because a large amount of data needed to be collected for a quantitative analysis. Recently, commercial instruments became available on the market. Another breakthrough was the integration of OT instruments with microfluidic systems, which accelerated both experimental setup and data acquisition (14,15). Nowadays, high-frequency data acquisition allows the generation of large data sets in a relatively short time. Subsequent data analysis, however, still requires custom written scripts to perform data preprocessing, identification of (un)folding events or different folding states, mathematical modeling, and statistical analysis. There are few algorithms developed for the analysis of single-molecule force spectroscopy data, which can perform alignment and pattern-recognition functions (20–23). Such algorithms are mostly tailored for atomic force spectroscopy data analysis and thus are not directly applicable for OT data (20–25). In addition, device manufacturers would provide basic solutions for the analysis of force spectroscopy data, yet processing of the data still require bioinformatics and statistics skills, and this therefore remains a major bottleneck.

Here, we present an automated python-based pipeline for the analysis of OT force-ramp and constant-force data (POTATO). Using statistical analysis of the time derivative of force and distance data, both unfolding as well as refolding steps are deduced automatically, and values such as (un)folding force and step length are derived. These values are then directly employed for fitting of force-distance (FD) curves. Additionally, we provide a basic constant-force analysis function. In order to allow the users to modify the

analysis parameters to suit their needs, we integrated an easy-to-use graphical user interface (GUI) in POTATO. Since the pipeline allows automated processing of multiple raw data files, our tool reduces the analysis time substantially, and the automated analysis ensures reproducibility and eliminates inconsistencies of manual analysis (26). Next, applicability of the tool is demonstrated on an artificially generated data set, which covers a broad range of possible parameter combinations for force-ramp data, and also on real experimental data (27,28). Finally, we also evaluated the performance of POTATO on a published data set independently generated using a self-built OT system (29). Our results indicate that POTATO exhibits a robust performance in identifying (un)folding events with high accuracy, precision, and recall.

MATERIALS AND METHODS

Algorithm implementation

The algorithm is written in python 3. We designed a GUI and wrapped the code into a Windows standalone executable with *pyinstaller* to open this tool to a broader audience without a bioinformatics background. The code is freely available on GitHub (https://github.com/REMI-HIRI/POTATO), and the architecture of the python files and GUI is further explained in the supporting material.

Artificial data generation

Artificial force spectroscopy data were generated using a custom-written python script (supporting material). The fully folded part of FD curves was modeled using an equation for extensible worm-like chain (WLC) models (Eq. 4). The partially unfolded region was modeled using a combination of WLC and freely jointed chain (FJC) models (Eqs. 5 and 6). For a more detailed description, see the supporting material.

Optical trapping system

OT experiments were performed using a C-Trap instrument (Lumicks, Amsterdam, the Netherlands). This device offers two laser traps combined with a 5-channel laminar-flow microfluidics system and a confocal microscope. Experiments were conducted as described in (27,28,30).

RESULTS AND DISCUSSION

Data preprocessing

Raw data (Fig. 1 *B*) from various input file formats (.h5 or .csv files containing force and distance information) can be loaded and preprocessed before marking the (un)folding events (supporting material). Depending on the data collection frequency, downsampling can be performed, which accelerates the analysis and saves storage space. Downsampling is especially crucial when data are collected at high frequencies. The instrument we used automatically collects data in the high-frequency mode (78,000 Hz), and the raw data need to be downsampled for ease of analysis. On the other hand, self-built systems allow collecting the data at lower frequencies. In principle, if the data frequency is sufficiently high to detect the molecule while transitioning from folded to unfolded states, and vice versa, POTATO can perform the analysis. Therefore, the downsampling rate should be defined by the user empirically. We also note that data sets of very low data-gathering frequency may not be suitable for direct analysis by POTATO. In that case, further preprocessing steps can be implemented (see data augmentation in supporting material). At the next step, a low pass Butterworth filter is employed to reduce the noise out of the signal (Eq. 1) (31). This filter allows efficient noise removal while keeping the actual (un)folding events intact and is therefore commonly used (Fig. 1 *C*). The algorithm then trims the data at a minimum force threshold set by the user (Table S1). Similar to downsampling, the noise filtering can also be disabled in the GUI if the loaded data is already preprocessed.

Butterworth filter:

$$G^2(\omega) = \frac{G_0^2}{1 + \left(\frac{\omega}{\omega}\right)^{2n}} \tag{1}$$

G is gain, ω is frequency, ω_c is cut-off frequency, and n is filter degree.

Force-ramp data analysis

For the identification of (un)folding events, we employed a derivative-based approach, which has been previously demonstrated to allow efficient step recognition (23). There are also other algorithms available that are based on probabilistic approaches, such as FEATHER (22). However, it must be noted that these tools are mostly developed for the analysis of atomic-force-microscopy-generated data (20-25). Here, we aimed to combine step recognition with downstream data fitting and determination of work, based solely on recorded force and distance values. Furthermore, we aimed to keep the pipeline intuitive and adjustable to user requirements. Although this tool was initially developed for the analysis of Lumicks FD data in H5 format, in principle, POTATO can be employed to analyze any data set format independent of the type of OT instrument.

Statistical analysis

In force-ramp trajectories, an unfolding event is characterized by a simultaneous drop in force and a quick increase in distance as the secondary structure of the polymer undergoes a sudden transition from the folded to the unfolded state (Fig. 1 *C*). Refolding events have opposite characteristics, in which the distance decreases and the force increases upon refolding. When flipped, the refolding data cannot be distinguished from the unfolding data and the processing, therefore step identification can be performed in an identical manner. Ultimately, these (un)folding events can be identified as a local maximum in the derivative of the distance

and a local minimum in the derivative of the force (Eq. 2). When plotted, the numerical derivative data of both distance and force show two populations of values. The first is a normal-like distribution representing the measurement noise, while outliers from the normal distribution represent the second population—the actual (un)folding events. To distinguish real (un)folding events from background noise, we calculate the moving median and the standard deviation (SD). These are then used to separate the normally distributed data from the extreme values outside a given Z score (i.e., number of SDs = 3 by default) (Fig. 1 *D*). This should include 99.73% of the normally distributed data points. As the initially calculated SD is affected by the outliers, a second SD is calculated from the data points inside the threshold, and the data are sorted again. The cycle is repeated until the difference between initial and secondary SD is <x (with x default = 5%). After the force and distance derivatives are sorted, our algorithm finds the local extrema of the derivatives, representing the saddle points of the (un) folding events in the FD curve. Then, it finds the adjacent crossing points of the derivative with the moving median, representing the start or end of the corresponding unfolding events.

Numerical approximation of the derivatives:

$$\frac{dF}{dt} = \frac{F(t+dt)-F(t)}{dt} \approx \lim_{\Delta t \to 0} \frac{F(t+dt)-F(t)}{dt} = \frac{F(x+step\ d)-F(x)}{step\ d}$$

$$\frac{dD}{dt} = \frac{D(t+dt)-D(t)}{dt} \approx \lim_{\Delta t \to 0} \frac{D(t+dt)-D(t)}{dt} = \frac{D(x+step\ d)-D(x)}{step\ d} \quad (2)$$

F is force, D is distance, t is time, x is position, and step d is a change in position.

Data fitting

Once the respective (un)folding steps are identified, this information is employed for data fitting. Data fitting is performed on the untrimmed data to model the trajectories more precisely. For the characterization of the mechanical properties of the (bio)polymer under tension, the extensible WLC model is commonly used, relating the applied force and molecular extension (Eq. 3) (32). For that, the FD curve is split into multiple parts. The fully folded part (until the first detectable unfolding step) is fitted with a WLC (32) to calculate the persistence length (dsL_P) of the tethered molecule, while the contour length (dsL_C) is fixed. In addition, baseline and offsets in both force and distance are included in the model to compensate for the experimental variability in the FD curves.

The partially and fully unfolded parts of the FD curves are subsequently fitted using a combined model comprising WLC (describing the folded double-stranded handles) and FJC (Eqs. 4 and 5) or another WLC model (representing the unfolded single-stranded parts) (Eq. 6) (Fig. 1 *E*) (32,33). To mathematically fit the models, we applied model polymer stretching functions from the free python package *pylake* (Lumicks).

Extensible WLC model:

$$x_{WLC} = L_C \left[1 - \frac{1}{2} \left(\frac{k_B T}{(F - F_{offset}) \cdot L_P} \right)^{1/2} + \frac{(F - F_{offset})}{K_0} \right] - d_{offset} \quad (3)$$

X is an extension, L_C is contour length, F is force, L_P is persistence length, k_B is Boltzmann constant, T is thermodynamic temperature, K_0 is stretch modulus, F_{offset} is force offset, and d_{offset} is distance offset.

FJC:

$$x_{FJC} = L_C \left[\coth\left(\frac{2F \cdot L_P}{k_B T} \right) - \frac{k_B T}{2F \cdot L_P} \right] \left(1 + \frac{F}{K_0} \right) \quad (4)$$

$WLC + FJC$:

$$x_{total} = x_{ds} + x_{ss} = x_{WLC} + x_{FJC} \quad (5)$$

$WLC + WLC$:

$$x_{total} = x_{ds} + x_{ss} = x_{WLC1} + x_{WLC2} \quad (6)$$

Work calculations

Unfolding and refolding FD trajectories also yield crucial information on the thermodynamic properties of the molecule under study. Accordingly, the work applied by the OT instrument onto the system can be calculated from the area under the FD curve (AUC), here using composite Simpson's rule (Eq. 7). First, we determine the work applied to the whole construct, including the handles (Fig. 2 *A*). The total work on the construct is the sum of the AUC of the folded model until the starting point of the step (W_{ds}) and work performed during the step transition (W_{step}), represented by the rectangular area of the step length times force average (($F_{start} + F_{end}$)/2) (Fig. 2 *A*). In order to extract the amount of work applied only to the structure of interest ($W_{structure}$; Fig. 2 *C*), the work applied to the handles, represented by the AUC of the combined model (W_{ss}), is subtracted from the sum of the work on the whole construct (Eq. 8; Fig. 2 *B* and *C*). It shall be noted that the work derived from these calculations equals the Gibbs free energy of the studied structure provided the system is in thermodynamic equilibrium. However, if the (un)folding trajectories do not coincide, it indicates that the molecule is out of equilibrium. In non-equilibrium scenario, Gibbs free energy can be extracted from the work values (5,18,19,29,34-36) (Fig. S3). It should be noted that while POTATO performs

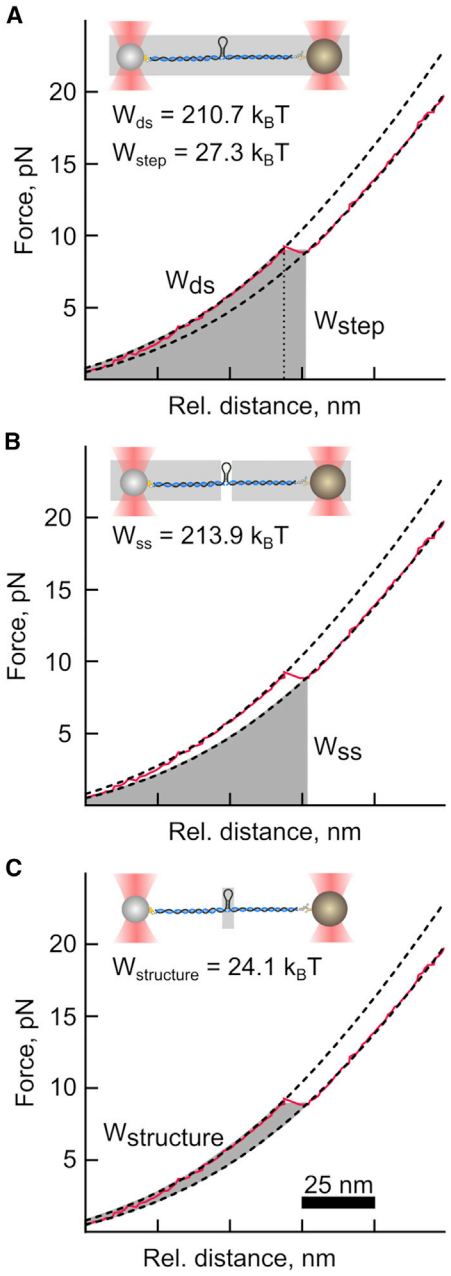

FIGURE 2 Work determination of a simple hairpin. (*A–C*) FD curve obtained during force-ramp experiment of a short stem loop of 30 nucleotides. Inlets: the optical tweezers construct stretched between the beads with gray regions indicating to what parts of the construct the calculated work relates. (*A*) Marked region (*gray*) corresponding to the work necessary for stretching of the whole construct including the structure of interest. (*B*) Marked region (*gray*) corresponding to the work necessary for stretching of the handles and the unfolded single-stranded RNA. (*C*) Marked region (*gray*) corresponding to the work necessary for stretching of the RNA structure of interest. See the subsequent analysis in Fig. S3. To see this figure in color, go online.

work calculations, the estimations of free-energy values have to be derived by the user separately.

Numerical integration using composite Simpson's rule:

$$\int_a^b f(x)dx \approx \frac{h}{3}\sum_{j=1}^{n/2}\left[f\left(x_{2j-2}\right)+4f\left(x_{2j-1}\right)+f\left(x_{2j}\right)\right], \quad (7)$$

where $x_j = a + jh \; for \; j = 0, 1, \ldots, n\text{-}1$ with $h=(b\text{-}a)/n$; $x_0 = a$ and $x_n = b$.

Non-equilibrium work calculation:

$$W_{structure} = W_{ds} + W_{step} - W_{ss} \quad (8)$$

$W_{structure}$ is work needed to unfold the structure of interest. W_{ds} is numerical integration of the fully folded model, W_{ss} is numerical integration of the unfolded model, and W_{step} is numerical integration of the step region between the two models.

Constant-force data analysis

In addition to force-ramp experiments, the algorithm we provide can also analyze constant-force data (Fig. S1). In this way, the dynamics of the structure at a given force can be investigated. This way, the equilibrium force at which the chance of the structure to be folded or unfolded are equal can be derived.

The constant-force analysis accepts the same input formats as the force-ramp batch analysis, and data preprocessing is performed similarly by downsampling and filtering of the data without trimming. First, it is necessary to display the constant-force data in order to optimize the preprocessing parameters and the plot's axis (Fig. S1 *B*). At this step, two plots are generated for visualization. In the first plot, distance is plotted against time. Here, the difference in distance corresponds to the change in the contour length of the tethered molecule. The second plot is a histogram of the distance distribution (Fig. S1 *C*). From this histogram, the number of different folding states can be deduced. Afterward, the histogram is fitted with multiple Gaussian functions. According to the position distribution histograms, the user can interactively provide initial estimates for various parameters including the number, localization, width (SD, *Z* score), and amplitude of the fits. After the optimization, the model parameters are exported together with the percentage of each folding state as a table in csv format (comma separated values).

Artificial data sets to test the limits of detection

To test the limits of (un)folding events detectable by the POTATO pipeline, an artificial data set was generated (supporting material). In this data set, some curves can show a negative step length that would not be observed in real unfolding events. We considered these steps as non-identifiable and used them as negative controls. The phenomenon of negative steps can mainly be observed for small contour-length changes (ΔL_C) between the models, combined

with high force drop (ΔF) values. To test the performance of the algorithm, we defined identifiable steps as events with a drop in force and a simultaneous increase in distance (supporting material). To evaluate if a specific parameter combination results in an identifiable curve, Eq. 9 with $x = 0$ was solved for all sets of parameters. Each time two parameters were fixed, and the third parameter was optimized.

Minimal step calculation:

$$x = \mathrm{WLC}_{ss}(\mathrm{step}_{\mathrm{end}}) + \mathrm{WLC}_{ds}(\mathrm{step}_{\mathrm{end}}) - \mathrm{WLC}_{ds}(\mathrm{step}_{\mathrm{start}}), \quad (9)$$

where WLC corresponds to expression from Eq. 3, *ss* refers to the model corresponding to single-strand values, and *ds* describes the double-stranded region.

A hyperplane showing the interface of theoretically identifiable and non-identifiable steps was generated from these optimized values (Fig. 3 *A*). This allowed us to classify the generated data set based on a combination of parameters: one with curves where POTATO is expected to find an unfolding step ($x > 0$) and the other one where POTATO should not identify the steps ($x \leq 0$). After analyzing the artificial data set (comprising 2520 curves) with different *Z* scores, the expected results, based on the input parameters when the data were generated, were compared with the steps identified by POTATO. For the default *Z* score of 3, the expected parameters were then plotted into the three-dimensional plot and colored based on the identification by POTATO (Fig. 3 *A*). For an unfolding force of 25 pN, the ΔF and ΔL_C values are shown in a two-dimensional plot, making it easier to identify and compare single unfolding events analyzed with different *Z* scores. It can be seen that all identified steps at this specific unfolding force are above the theoretical threshold and that more unfolding events are identified at *Z* score 2.5 than at 3 (Fig. 3 *B*). Accordingly, the effect of the *Z* score on the derivative of force (Fig. 3 *C*) and distance (Fig. 3 *D*) can be investigated for an individual FD trajectory. In the representative trajectory, the local maximum in the derivatives of distance is above the *Z* score threshold for both cases. In the derivative of force, the local minimum at the same position is only detected for the lower *Z* score (Fig. 3 *C* and *D*).

Next, we calculated performance measures such as accuracy, precision, sensitivity, specificity, and F1 score to validate the performance of POTATO. For a *Z* score of 3.2, a precision score of 0.974 indicates that most of the positive classified steps were actual steps, and even for a *Z* score of 2.5, the precision was still above 0.944 (Table S2). As expected, higher precision comes with the trade-off to miss certain positive events (recall 0.870–0.939), and the optimal *Z* score has to be chosen depending on the application. For smaller unfolding events that are difficult to detect, lower *Z* score should be employed, as for distinct unfolding events, the *Z* score can be set to higher values. This way, the number of false-positive events detected can be minimized. Since the present data set was generated using artificial parameter combinations, those might not be found in actual OT measurements. Therefore, it is important to keep in mind that we were exploring the limits of the tool by using these strict parameter constraints. Performance measures would also vary depending on where a specific data set is located in the parameter space and which *Z* scores were employed.

Furthermore, we investigated how accurately POTATO estimates step parameters (F_U, ΔL_C, ΔF). For that, we compared the expected and measured values of these parameters for all curves analyzed (Fig. 4). We then calculated the linear regression of the true positive values to estimate possible biases of POTATO-estimated F_U and ΔL_C values. Our analysis shows that in the case of F_U (Fig. 4 *A*), the values determined by POTATO are in perfect agreement with the expected values (slope of the linear regression = 0.9912). For ΔL_C (Fig. 4 *B*), the comparison shows a broader distribution of the measured values, with an overall trend suggesting a minor overestimation (slope of the linear regression = 1.0282) of around 3%. Lastly, in the case of ΔF (Fig. 4 *C*), the trend shows a slight underestimation of the measured values (slope of the linear regression = 0.8517), resulting in a bias of 12%–15%. Taken together, our performance-measures analysis suggests that the presented tool successfully identifies most (un)folding events correctly with only few false classifications (false positives/false negatives). Accordingly, in most of the cases, performance measures were above 0.9 (Table S2). Moreover, we show that POTATO can precisely estimate the parameter values describing the (un)folding events (F_U, ΔL_C, ΔF; Fig. 4). Overall, the performance measures and the accuracy of the estimates show that POTATO represents a reliable tool for optical tweezer data analysis.

Applicability of POTATO on real experimental data

Next, we employed POTATO to test its performance on real experimental data generated from FD measurements of the programmed ribosomal frameshifting element of the encephalomyocarditis virus and severe acute respiratory syndrome coronavirus 2 (27,28). We compared the POTATO results with manually annotated steps of a subset of our data set. The results obtained with manual step identification and data fitting were in good agreement with the automated analysis using the pipeline (Fig. S2 *A*). Harnessing POTATO in the data processing allowed us to speed up the analysis significantly compared with previous manual analysis. Furthermore, we saw that POTATO is not only suitable for curves with a single (un)folding event like in the artificial data set, but we successfully fit FD curves with as many as five unfolding steps, and we were able to identify even short-lived intermediate states of the unfolding process (Fig. S2 *B* and *C*). In addition to the contour-length

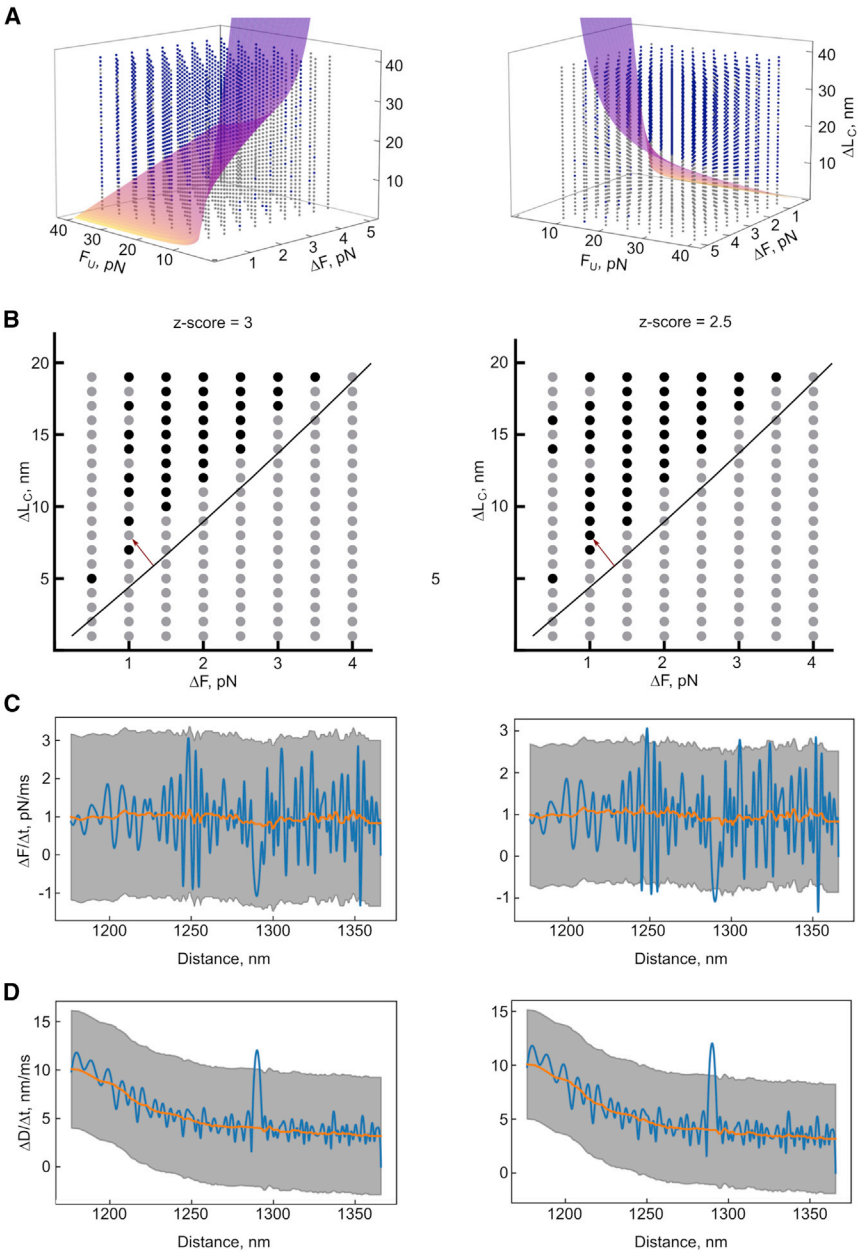

FIGURE 3 Testing the limits of POTATO. For each combination of the parameters unfolding force (F_U), force drop (ΔF), and contour-length change (L_C), two parameters were fixed, and the third one was optimized so that the Eq. 9 (supporting material) evaluates to zero. (*A*) A hyperplane was generated from the optimized values that separate the resolvable space above the hyperplane (parameter combinations that result in identifiable steps) from the unresolvable space below the hyperplane (parameter combinations that result in unidentifiable steps). Each analyzed curve is plotted in blue if its step was identified by POTATO or in gray if it was not recognized. (*B*) Slices of the three-dimensional plot at $F_U = 25$ pN were analyzed with different *Z* scores. The black line corresponds to the theoretical limit of resolvable/unresolvable parameter combinations. The black dots represent curves with identified steps, whereas the gray dots represent curves where POTATO could not identify the step. (*C* and *D*) The derivatives of force (*C*) and distance (*D*) of the curve that is marked with a red arrow in (*B*) are displayed at different *Z* scores.

change obtained by curve fitting, the Gibbs free energy is also an important variable to conclude on the nature of the (un)folded structure as it is dependent on the base pairing of the RNA. We were able to use the work calculated by the POTATO to estimate the Gibbs free energy of the structures and thereby distinguish between different secondary structures (27). Here, to demonstrate the energy calculation, we used a stem-loop mRNA of 30 nucleotides in length (Fig. S3) (28). First, we used mfold (37) to predict the secondary structure and its Gibbs free energy (Fig. S3 *A*). Then, we plotted the unfolding as well as refolding work distributions calculated by POTATO (Fig. S3 *B*). We then employed the results of POTATO analysis to estimate the Gibbs free energies by applying 1) Crooks fluctuation theorem and 2) Jarzynski equality with bias correction (Fig. S3 *C*) as described in (18,34–36).

To evaluate the performance of POTATO on other published data sets generated using a self-built OT instrument, we analyzed the severe acute respiratory syndrome coronavirus 2 pseudoknot RNA FD data by Neupane et al. (29). Since the data set provided had a lower data frequency, resulting in less than 250 data points per FD curve, we first had to artificially augment the datapoints (see supporting material). Despite that, we could still successfully assign the steps and reproduce the unfolding force distribution (Fig S2) as well as the contour-length estimate (Table S3).

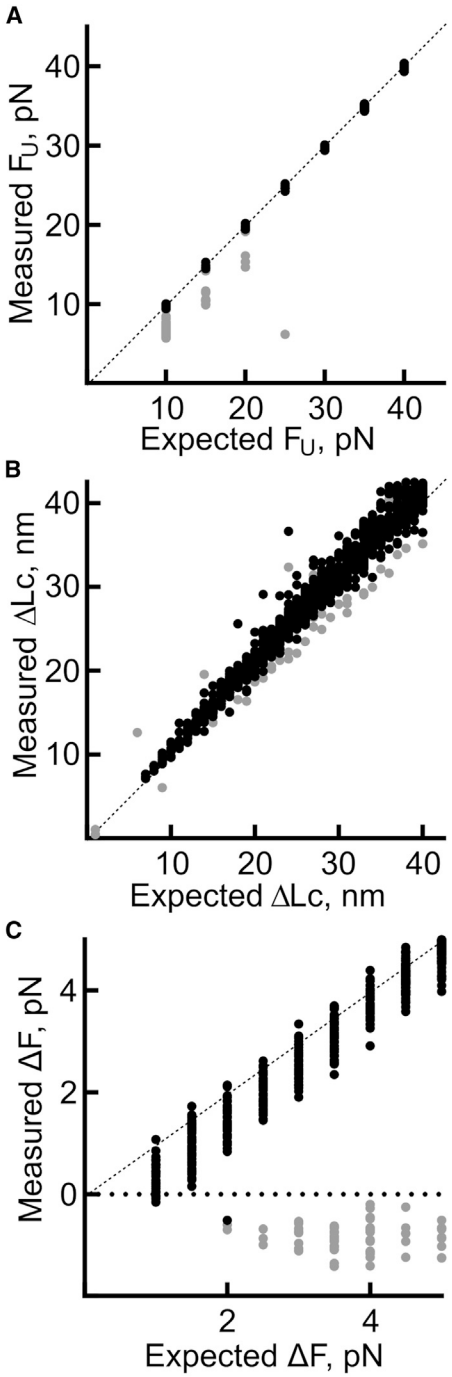

FIGURE 4 Evaluation of the performance of POTATO. The parameters used for the generation of the data set compared with the parameters identified by POTATO are plotted against each other. All three parameters used for the data generation are evaluated with a *Z* score of 3. (*A–C*) The values of the true positive steps (*black*) and the values of the false-positive steps (*gray*) are visualized for (*A*) the unfolding force (F_U), (*B*) the contour length change (ΔL_C), and (*C*) the force drop (ΔF). A dashed line represents the theoretical perfect correlation between measured and expected value.

We were also able to detect the refolding steps' force distribution and detected steps as low as 6 pN (Fig. S2). In conclusion, regardless of the system used, we demonstrate that the pipeline output matched well with manual data analysis on real-experiment data sets and that POTATO performed analysis of FD trajectories with multiple steps or even short-live intermediates in a reliable way. Therefore, POTATO represents a versatile tool for high-throughput OT data analysis for many upcoming studies.

Limitations of the study

Processing automation comes with trade-offs (38,39). First, the statistical analysis applied in the pipeline might be prone to false-positive event discoveries due to external causes, such as vibration that might induce step-like events in the FD profile of gathered data. We split the FD data and analyze the derivatives of force and distance separately to minimize this effect. Only the events found by both approaches are considered real (un)folding events. Therefore, the robustness of the analysis is increased.

Second, the pipeline output strongly depends on parameters and threshold values that are applied throughout the analysis. The default values were set empirically to suit our needs. Therefore, it might require optimization to fit specific needs and reach an analysis output consistent with the manual data analysis. User input is required despite the user-friendly GUI environment, and an understanding of the analysis workflow is necessary to adjust the parameters rationally.

The current algorithm does not annotate the repeated folding and unfolding of a structure during force-ramp measurements and identifies this oscillation as independent steps. Nevertheless, this mainly occurs at slow loading rates and does not affect the contour-length estimates. To overcome any unexpected issues with the automated analysis, POTATO also includes a tab that allows full manual analysis of the force-ram data files. This should help to eliminate bias caused by omission of certain files from the analysis during the automated analysis.

Summary

Here, we present a publicly available pipeline for batch analysis of OT data. Our pipeline allows OT raw or preprocessed data processing from force-ramp or equilibrium measurements (constant force/position). These are widely employed experimental approaches in the OT field, applied to nucleic acid structure probing, protein folding, RNA-protein interactions, or even to analyze events as complex as translation. Here, by wrapping our algorithm in a standalone application and designing an intuitive GUI, we aim to open the data analysis to a broader audience without the need for a bioinformatics background. The user can adjust all parameters directly in the GUI without diving into the code to tailor the pipeline to their exact needs. With the parameters optimized for the here-presented data sets, POTATO showed high precision and accuracy in the identification of (un)

folding events. Moreover, compared with manual data analysis, the pipeline is faster and, most importantly, consistent throughout the analysis, thus yielding reproducible results.

SUPPORTING CITATIONS

References (40–44) appear in the supporting material.

SUPPORTING MATERIAL

Supporting material can be found online at https://doi.org/10.1016/j.bpj.2022.06.030.

AUTHOR CONTRIBUTIONS

N.C., L.P., and S.B. designed the pipeline. L.P. and S.B. wrote the python scripts. L.P. generated the artificial data. S.B. analyzed the artificial data. L.P. and S.B. performed the OT experiments. L.P. analyzed experimental data. L.P. and S.B. prepared the figures with input from N.C. N.C., L.P., and S.B. wrote the manuscript.

ACKNOWLEDGMENTS

We thank Vojtech Vrba for helpful python discussions. We thank Dr. Anke Sparmann for critically reviewing the manuscript. The work in our laboratory is supported by the Helmholtz Association and European Research Council (ERC) grant no. 948636.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

1. Ashkin, A., J. M. Dziedzic, …, S. Chu. 1986. Observation of a single-beam gradient force optical trap for dielectric particles. *Opt. Lett.* 11:288. https://doi.org/10.1364/OL.11.000288.
2. Moffitt, J. R., Y. R. Chemla, …, C. Bustamante. 2008. Recent advances in optical tweezers. *Annu. Rev. Biochem.* 77:205–228. https://doi.org/10.1146/annurev.biochem.77.043007.090225.
3. Choudhary, D., A. Mossa, …, C. Cecconi. 2019. Bio-molecular applications of recent developments in optical tweezers. *Biomolecules.* 9:23. https://doi.org/10.3390/biom9010023.
4. Hashemi Shabestari, M., A. E. C. Meijering, and E. J. G. Peterman. 2017. Chapter four - recent advances in biological single-molecule applications of optical tweezers and fluorescence microscopy. *In* Methods Enzymol. M. Spies and Y. R. Chemla, eds. Academic Press, pp. 85–119.
5. Bustamante, C. J., Y. R. Chemla, …, M. D. Wang. 2021. Optical tweezers in single-molecule biophysics. *Nat. Rev. Methods Primers.* 1:25. https://doi.org/10.1038/s43586-021-00021-6.
6. Chen, Y.-T., K.-C. Chang, …, J. D. Wen. 2017. Coordination among tertiary base pairs results in an efficient frameshift-stimulating RNA pseudoknot. *Nucleic Acids Res.* 45:6011–6022. https://doi.org/10.1093/nar/gkx134.
7. Mukhortava, A., M. Pöge, …, M. Schlierf. 2019. Structural heterogeneity of attC integron recombination sites revealed by optical tweezers. *Nucleic Acids Res.* 47:1861–1870. https://doi.org/10.1093/nar/gky1258.
8. Stephenson, W., G. Wan, …, P. T. X. Li. 2014. Nanomanipulation of single RNA molecules by optical tweezers. *JoVE.* 90. https://doi.org/10.3791/51542.
9. Zhong, Z., L. Yang, …, G. Chen. 2016. Mechanical unfolding kinetics of the SRV-1 gag-pro mRNA pseudoknot: possible implications for -1 ribosomal frameshifting stimulation. *Sci. Rep.* 6:39549. https://doi.org/10.1038/srep39549.
10. Jiao, J., A. A. Rebane, …, Y. Zhang. 2017. Single-molecule protein folding experiments using high-precision optical tweezers. *Methods Mol. Biol.* 1486:357–390. https://doi.org/10.1007/978-1-4939-6421-5_14.
11. Ritchie, D. B., J. Soong, …, M. T. Woodside. 2014. Anti-frameshifting ligand reduces the conformational plasticity of the SARS virus pseudoknot. *J. Am. Chem. Soc.* 136:2196–2199. https://doi.org/10.1021/ja410344b.
12. Desai, V. P., F. Frank, …, C. Bustamante. 2019. Co-temporal force and fluorescence measurements reveal a ribosomal gear shift mechanism of translation regulation by structured mRNAs. *Mol. Cell.* 75:1007–1019.e5. https://doi.org/10.1016/j.molcel.2019.07.024.
13. Liu, T., A. Kaplan, …, C. J. Bustamante. 2014. Direct measurement of the mechanical work during translocation by the ribosome. *Elife.* 3:e03406. https://doi.org/10.7554/eLife.03406.
14. Eriksson, E., J. Enger, …, D. Hanstorp. 2007. A microfluidic system in combination with optical tweezers for analyzing rapid and reversible cytological alterations in single cells upon environmental changes. *Lab Chip.* 7:71–76. https://doi.org/10.1039/B613650H.
15. Gross, P., G. Farge, …, G. J. L. Wuite. 2010. Combining optical tweezers, single-molecule fluorescence microscopy, and microfluidics for studies of DNA-protein interactions. *Methods Enzymol.* 475:427–453. https://doi.org/10.1016/s0076-6879(10)75017-5.
16. Whitley, K. D., M. J. Comstock, and Y. R. Chemla. 2017. High-resolution "fleezers": dual-trap optical tweezers combined with single-molecule fluorescence detection. *Methods Mol. Biol.* 1486:183–256. https://doi.org/10.1007/978-1-4939-6421-5_8.
17. Rocha, M. S. 2009. Optical tweezers for undergraduates: theoretical analysis and experiments. *Am. J. Phys.* 77:704–712. https://doi.org/10.1119/1.3138698.
18. McCauley, M. J., I. Rouzina, …, M. C. Williams. 2020. Significant differences in RNA structure destabilization by HIV-1 GagΔp6 and NCp7 proteins. *Viruses.* 12:484. https://doi.org/10.3390/v12050484.
19. McCauley, M. J., I. Rouzina, …, M. C. Williams. 2015. Targeted binding of nucleocapsid protein transforms the folding landscape of HIV-1 TAR RNA. *Proc. Natl. Acad. Sci. USA.* 112:13555–13560. https://doi.org/10.1073/pnas.1510100112.
20. Kuhn, M., H. Janovjak, …, D. J. Muller. 2005. Automated alignment and pattern recognition of single-molecule force spectroscopy data. *J. Microsc.* 218:125–132. https://doi.org/10.1111/j.1365-2818.2005.01478.x.
21. Bosshart, P., P. Frederix, and A. Engel. 2012. Reference-free alignment and sorting of single-molecule force spectroscopy data. *Biophys. J.* 102:2202–2211. https://doi.org/10.1016/j.bpj.2012.03.027.
22. Heenan, P. R., and T. T. Perkins. 2018. FEATHER: automated analysis of force spectroscopy unbinding and unfolding data via a Bayesian algorithm. *Biophys. J.* 115:757–762. https://doi.org/10.1016/j.bpj.2018.07.031.
23. Andreopoulos, B., and D. Labudde. 2011. Efficient unfolding pattern recognition in single molecule force spectroscopy data. *Algorithm Mol. Biol.* 6:16. https://doi.org/10.1186/1748-7188-6-16.
24. Gergely, C., B. Senger, …, J. Hemmerlé. 2001. Semi-automatized processing of AFM force-spectroscopy data. *Ultramicroscopy.* 87:67–78. https://doi.org/10.1016/s0304-3991(00)00063-2.
25. Roduit, C., B. Saha, …, S. Kasas. 2012. OpenFovea: open-source AFM data processing software. *Nat. Methods.* 9:774–775. https://doi.org/10.1038/nmeth.2112.
26. Muhs, K. S., W. Karwowski, and D. Kern. 2018. Temporal variability in human performance: a systematic literature review. *Int. J. Ind. Ergon.* 64:31–50. https://doi.org/10.1016/j.ergon.2017.10.002.

27. Hill, C. H., L. Pekarek, …, I. Brierley. 2021. Structural and molecular basis for Cardiovirus 2A protein as a viral gene expression switch. *Nat. Commun.* 12:7166. https://doi.org/10.1038/s41467-021-27400-7.

28. Zimmer, M. M., A. Kibe, …, N. Caliskan. 2021. The short isoform of the host antiviral protein ZAP acts as an inhibitor of SARS-CoV-2 programmed ribosomal frameshifting. *Nat. Commun.* 12:7193. https://doi.org/10.1038/s41467-021-27431-0.

29. Neupane, K., M. Zhao, …, M. T. Woodside. 2021. Structural dynamics of single SARS-CoV-2 pseudoknot molecules reveal topologically distinct conformers. *Nat. Commun.* 12:4749. https://doi.org/10.1038/s41467-021-25085-6.

30. Pekarek, L., S. Buck, and N. Caliskan. 2022. Optical tweezers to study RNA-protein interactions in translation regulation. *JoVE.* 180:e62589. https://doi.org/10.3791/62589.

31. Butterworth, S. 1930. On the theory of filter amplifiers. *Wireless Engineer.* 7:536–541.

32. Odijk, T. 1995. Stiff chains and filaments under tension. *Macromolecules.* 28:7016–7018. https://doi.org/10.1021/ma00124a044.

33. Smith, S. B., Y. Cui, and C. Bustamante. 1996. Overstretching B-DNA: the elastic response of individual double-stranded and single-stranded DNA molecules. *Science.* 271:795–799. https://doi.org/10.1126/science.271.5250.795.

34. Gore, J., F. Ritort, and C. Bustamante. 2003. Bias and error in estimates of equilibrium free-energy differences from nonequilibrium measurements. *Proc. Natl. Acad. Sci. USA.* 100:12564–12569. https://doi.org/10.1073/pnas.1635159100.

35. Liphardt, J., S. Dumont, …, C. Bustamante. 2002. Equilibrium information from nonequilibrium measurements in an experimental test of Jarzynski's equality. *Science.* 296:1832–1835. https://doi.org/10.1126/science.1071152.

36. Collin, D., F. Ritort, …, C. Bustamante. 2005. Verification of the Crooks fluctuation theorem and recovery of RNA folding free energies. *Nature.* 437:231–234. https://doi.org/10.1038/nature04061.

37. Zuker, M. 2003. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.* 31:3406–3415. https://doi.org/10.1093/nar/gkg595.

38. Alberdi, E., L. Strigini, and P. Ayton. 2009. Why are people's decisions sometimes worse with computer support? *In* Computer Safety, Reliability, and Security. B. Buth, G. Rabe, and T. Seyfarth, eds. Springer Berlin Heidelberg.

39. Cummings, M. L., F. Gao, and K. M. Thornburg. 2016. Boredom in the workplace: a new look at an old problem. *Hum. Factors.* 58:279–300. https://doi.org/10.1177/0018720815609503.

40. Harris, C. R., K. J. Millman, …, T. E. Oliphant. 2020. Array programming with NumPy. *Nature.* 585:357–362. https://doi.org/10.1038/s41586-020-2649-2.

41. Collette, A. 2013. Python and HDF5. O'Reilly.

42. McKinney, W. 2010. Data structures for statistical computing in python. *Proc. 9th Python Sci. Conf.* 445:51–56. https://doi.org/10.25080/Majora-92bf1922-00a.

43. Virtanen, P., R. Gommers, …, Y. Vázquez-Baeza. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods.* 17:261–272. https://doi.org/10.1038/s41592-019-0686-2.

44. Hunter, J. D. 2007. Matplotlib: a 2D graphics environment. *Comput. Sci. Eng.* 9:90–95. https://doi.org/10.1109/MCSE.2007.55.

Biophysical Journal, Volume *121*

Supplemental information

POTATO: Automated pipeline for batch analysis of optical tweezers data

Stefan Buck, Lukas Pekarek, and Neva Caliskan

SUPPORTING MATERIAL

POTATO: Automated pipeline for batch analysis of optical tweezers data

Stefan Buck [†1], Lukas Pekarek[†1], Neva Caliskan* [1,2]

[†] Authors contributed equally to this work.

* Corresponding author

[1] Helmholtz Institute for RNA-based Infection Research (HIRI), Würzburg, Germany

[2] Medical Faculty, Julius-Maximilians University Würzburg, Würzburg, Germany

Script structure

The script is written in Python 3 and split into multiple parts for clarity. The first part, "POTATO_GUI", defines the GUI with all necessary functions and input variables. When the GUI is started, the default values of the input variables are loaded from the "POTATO_config" file. The GUI was created and structured using the standard Tkinter python package. A parallel subprocess initiates from this main process when a folder is selected for force ramp analysis to perform computationally demanding data-processing. This way the GUI remains responsive during computation. All the functions used for data preprocessing and step recognition are defined in the "POTATO_preprocessing" and the "POTATO_find_steps" files respectively. The functions used for curve fitting are defined in another file, "POTATO_fitting". For computation, we mainly use matplotlib and NumPy packages, as well as the lumicks.pylake package for fitting (**Table S4**). The subprocess is a daemon process spawned by the main process and therefore stops as soon as the GUI terminates the main process. The last part, "POTATO_constantF", is executed by the main thread as it only analyzes one constant force file at a time. The results are exported in different CSV files or as PNG images.

Graphical user interface

We designed a graphical user interface (GUI) that allows users to easily adjust the analysis steps and parameters according to their needs and select between three different input data formats. This enables the GUI to load data from every OT instrument. The GUI is separated into multiple tabs, resulting in easy and intuitive navigation without overloading the individual windows. The "POTATO_config" file, included in the POTATO repository, contains the default parameters, which are loaded into the GUI. The most commonly changed parameters can be found in the first tab, “Folder Analysis”, so a basic analysis can be performed right away (press enter to confirm changed parameters). Alternatively, before each analysis, all parameters can be adjusted in the 'Advanced Settings' tab to suit the data set. In addition, we implemented the possibility to selectively export results. Each analysis creates a new folder with a timestamp directly in the analyzed directory. The used parameters are exported as well so that parameters can be optimized later. The second tab, “Show Single File”, provides a control mechanism for data preprocessing. A single file can be loaded, and the unfiltered data are plotted together with the filtered data, which streamlines troubleshooting. Finally, there is a third tab for “Constant Force Analysis”.

Input data

The presented pipeline accepts three different input data formats. Two of them are based on the default hdf5 output format of Lumicks C-Trap – one is predefined for high-frequency data (using the piezo-tracking function of the instrument), and the second is for low-frequency data (using video recognition). The third data format is a basic CSV file format with force and distance values in the first and second columns. Force data need to be in [pN], whereas the unit of distance data can be specified either as [µm] or [nm] in the GUI. Thus, our pipeline can process force-distance data from virtually any optical tweezers machine. In addition, entire directories containing force-ramp data files can be selected and processed simultaneously.

Data output

Depending on individual analysis requirements, different export settings can be selected. The down sampled and filtered data are exported in CSV format (smooth) for each file by default. The identified (un)folding steps by derivatives of force and distance are exported together with the steps identified by both strategies (common steps) into a single CSV file. All identified steps of all curves in the analyzed folder are gathered into a single results file for quantitative analysis. The respective summary figure containing the plot of preprocessed data, trimmed data, and both derivatives with

marked steps is exported. The plots of fitted data, together with the fitting parameters and model data, are exported as PNG and CSV files, respectively.

Artificial data generation

To test the limits of the algorithm, artificial data with a single step per curve were generated. The fully folded part of force-distance curves was modeled using an equation for extensible WLC models (**Eq. 4**). The partially unfolded region was modeled using a combination of WLC and FJC models (**Eq. 5 and 6**). The force value at which the step occurs, the contour length change between the unfolded and folded region, and the drop in force during the step, are the parameters for data generation. The first parameter was set to occur between 10-40 pN with a 5 pN resolution. The curves were generated with a contour length change from 1-40 nm with a 1 nm resolution and a force drop of 1-5 pN with a 0.5 pN resolution. To mimic the (Gaussian) noise affecting the raw data, we employed the NumPy random normal distribution function (1).

Since the (un)folding step is generally defined as a drop in force (one of the parameters) and a sudden increase in distance (not a parameter), the data generated by this script also contained combinations that did not increase distance. We used these curves showing no increase in the distance as negative controls.

Augmentation of low-frequency data

During analysis of the freely available data from Neupane and Zhao et al., 2021, we had to employ the data augmentation approach to increase the precision of the analysis. For the best output, ideally raw data should be directly curated in POTATO and at least >2000 data points are available. The augmentation was performed as follows. For each two consecutive data points in the original data, we divided the linear space between them by factor of 100 to get positions for new data points. Starting from the first original data point, we consecutively added 99 new data values always increasing by the previously calculated increment +/- randomly assigned noise in force and distance dimensions using random gauss function with the parameters mu=0 and sigma=0.5. The newly created files where then analyzed as csv files by POTATO.

Manual data analysis

POTATO GUI also contains a tab that provides the user with the option to manually mark steps, fit models and calculate the work for FD curves (Manual Analysis – TOMATO). Manual analysis is particularly useful to evaluate the precision of the automated analysis and perform parameter optimization. Furthermore, manual analysis is a convenient option for the analysis of FD curves which cannot be analyzed properly by the batch analysis. To speed up the manual analysis we implemented several keyboard shortcuts, which allow switching between FD curves under a given directory and marking steps with save, delete functions among others. For the manual analysis, the initial step is to mark the beginning and end of each (un)folding event. Afterward, similar to the batch analysis, different parts of the curves will be fitted automatically based on the parameters entered at the GUI. In addition, the work corresponding to each (un)folding event is calculated based on the fitting (**Eq. 8**). During manual analysis, certain parameter constraints and fitting parameters such as contour length, persistence length, stiffness, distance and force offsets can be defined. A detailed set of parameter constraints can also be found under the 'Advanced Settings' tab. For further description, we suggest the reader to refer to the readme file on Github (https://github.com/REMI-HIRI/POTATO).

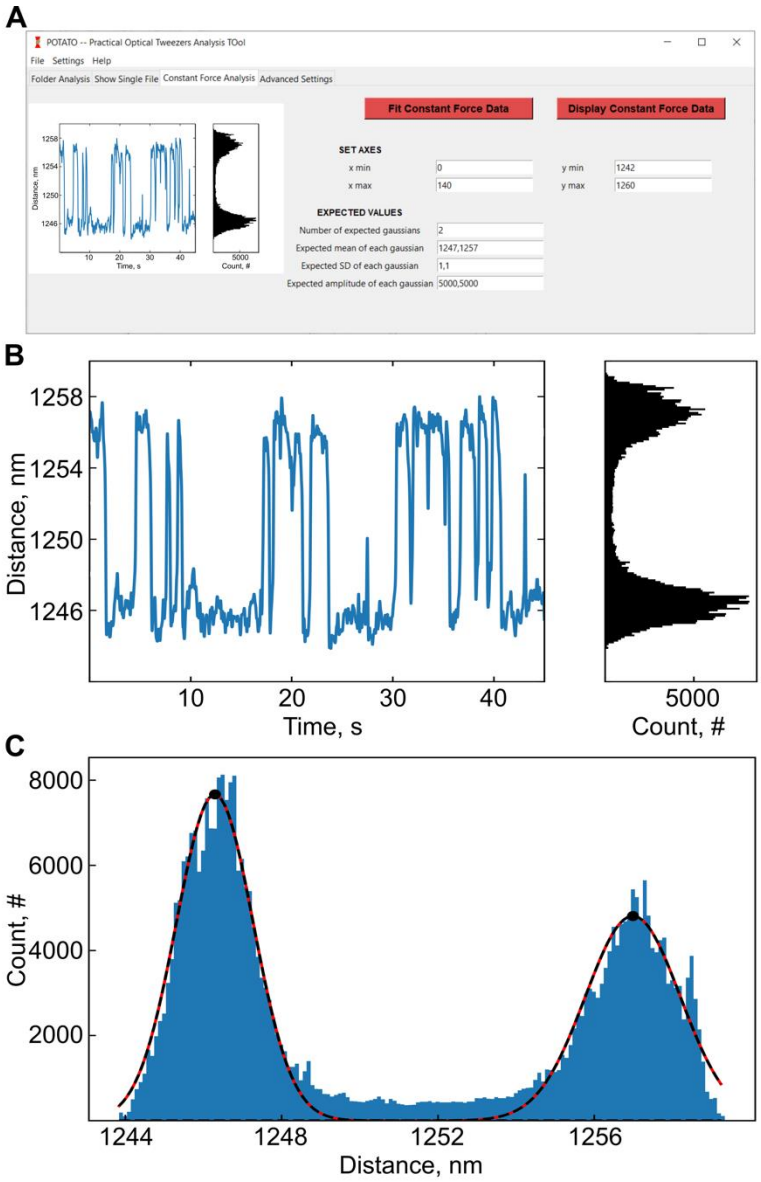

FIGURE S1: Constant force data analysis in POTATO: (A) GUI tab containing the constant force analysis features, **(B)** Display constant force data output; (left) distance over time plot, (right) histogram of the distance over time values. **(C)** Fit constant force data output showing the histogram of distance values distributions and the two gaussian functions fitted.

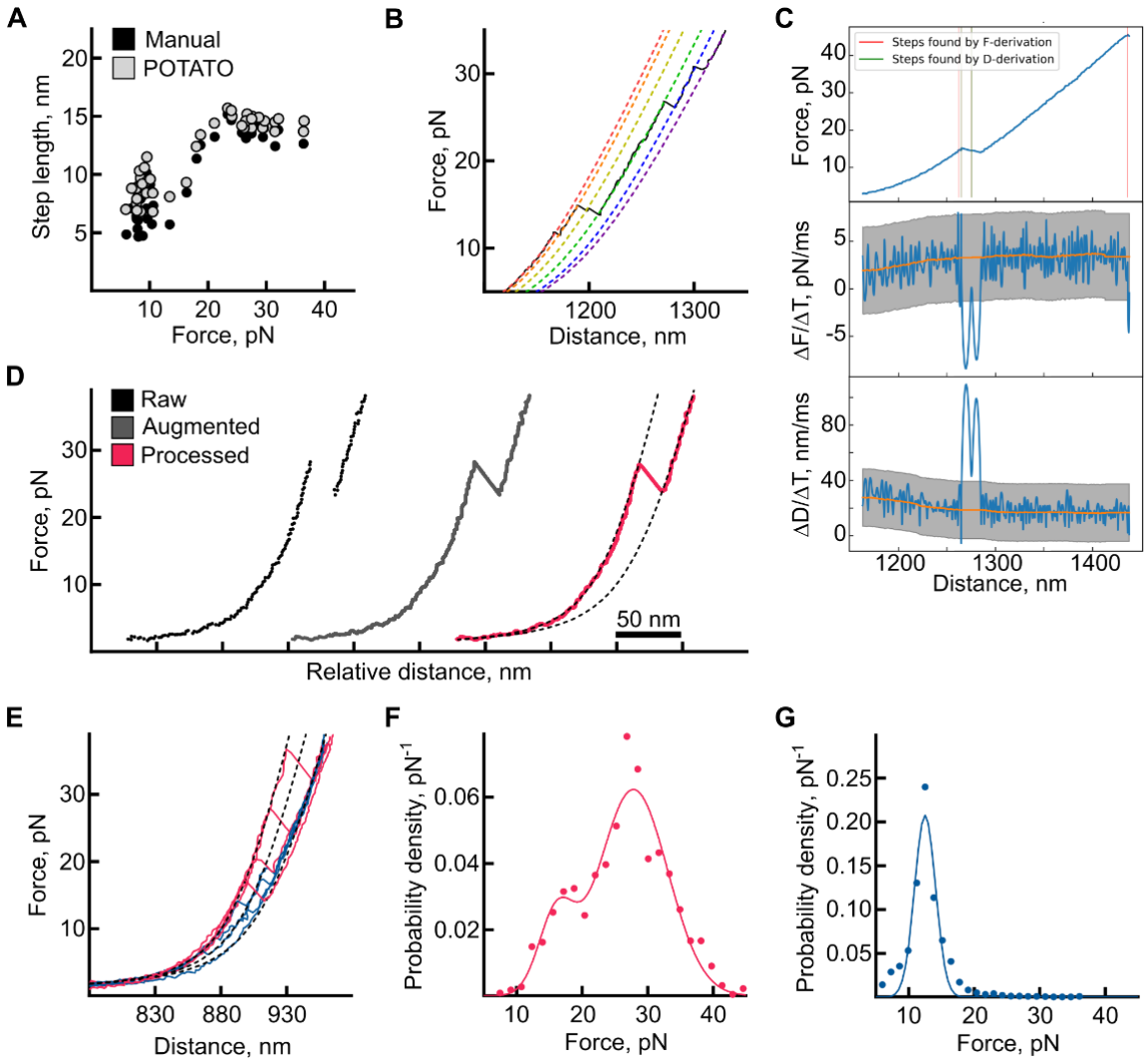

FIGURE S2: Analysis of experimental data by POTATO. **(A)** Comparison of unfolding events marked in a subset of the data analysed manually (black) or with POTATO (grey). (**B**) Example FD curve (black, solid) with five unfolding steps fitted by POTATO (colored, dashed). **(C)** Example analysis output from POTATO showing the trimmed FD curve (up), force derivative data (middle), and distance derivative data (bottom). An intermediate conformer is detected by POTATO during the unfolding. Other FD curves confirmed the presence of an even more stable and distinct intermediate step. (**D-G**) Analysis of experimental data published in Neupane and Zhao et al., 2021 (subset with 6nt spacer) using POTATO; (**D**) Comparison of raw data (black), data after augmentation (see also supplementary methods, grey), and data processed by POTATO (pink). **(E)** Example unfolding (pink) and refolding (blue) FD curves (n=4). (**F**) Distribution of unfolding forces for unfolding curves with single unfolding step (N=1378). (**G**) Refolding force distribution for all refolding curves (N=1861) shows a single peak around 12 pN with refolding steps detected at forces as low as 6 pN. (Un)folding distributions were overall similar to the analysis performed by Neupane and Zhao et al., 2021.

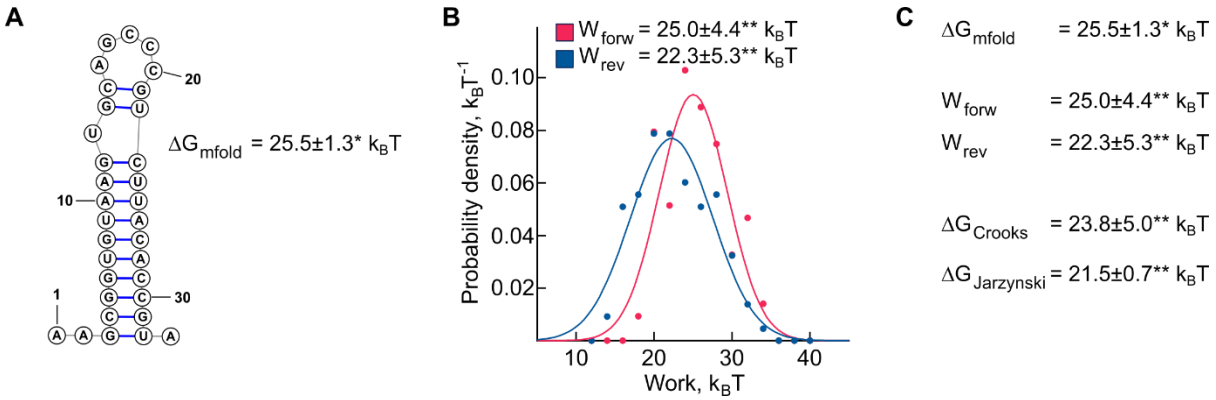

FIGURE S3: Extracting energy information from the experimental data. **(A)** Mfold predicted secondary structure of a simple hairpin of 30 nucleotides in lenght. **(B)** Distributions of measured work values for the unfolding (red) and refolding (blue) FD curves. **(C)** Energy and work values as predicted by Mfold (ΔG_{mfold}), measured (W_{forw} and W_{rev}) or calculated using Crooks Theorem (ΔG_{Crooks}) and Jarzynski equality ($\Delta G_{Jarzynski}$). * 5% standard error, **standard deviation.

TABLE S1: Parameters used throughout the pipeline and a short description.

Parameter	Description
Preprocessing	
Downsampling rate	Only every n^{th} value is taken for analysis, speeds up subsequent processing.
Butterworth filter degree	Defines the stringency of the filter.
Cut-off frequency	Signals with a frequency above this threshold are suppressed.
Force threshold, pN	Values lower than the threshold are excluded from the analysis.
Derivative	
Step d	Characterizes the interval between two values used for numerical derivative calculation.
Data frequency, Hz	The frequency at which data is recorded.
Statistics	
z-score	The number of standard deviation used to determine whether a given value is part of a normal distribution.
Moving median window size	The number of values considered for each median calculation.
SD difference threshold	Statistical analysis and data sorting are iterated until the difference between two consecutive SDs is below this value.
Fitting	
dsLp, nm	Persistence length of the double-stranded (folded) part of the tethered construct.
dsLc, nm	Contour length of double-stranded (folded) part of the tethered construct.
dsK0, pN	Stretch modulus of double-stranded (folded) part of the tethered construct.
Force offset, pN	Force offset of a given dataset; compensates for a shift in the dataset.
Distance offset, nm	Distance offset of a given dataset; compensates for a shift in the dataset.
ssLp, nm	Persistence length of the single-stranded (unfolded) part of the tethered construct.
ssLc, nm	Contour length of single-stranded (unfolded) part of the tethered construct.
ssK0, pN	Stretch modulus of single-stranded (unfolded) part of the tethered construct.

TABLE S2: Dependence of the performance measures on the z-score. Analysis of 2520 simulated data curves with steps occurring between 10-40 pN with different z-score values.

z-score / Parameter	3.2	3	2.7	2.5
True positives	1206	1267	1280	1303
True negatives	1101	1076	1073	1056
False positives	32	57	60	77
False negatives	181	120	107	84
Accuracy	0.915	0.930	0.934	0.936
Precision	0.974	0.957	0.955	0.944
Recall	0.870	0.913	0.923	0.939
Specificity	0.972	0.950	0.947	0.932
F1-Score	0.919	0.935	0.939	0.942

TABLE S3: Application of POTATO on experimental data generated from a simple hairpin and SARS-CoV-2 pseudoknot.

	Expected ΔLc, nm	Observed ΔLc, nm	Observed ΔLc, nm (Neupane et al. 2021)
Simple hairpin (30 nt)	17.7	16.4±2.8	-
SARS-CoV-2 frameshift pseudoknot (6 nt spacer)	34.7-36.3	34.8±2.0	35.6±0.4

TABLE S4: Python packages used in POTATO. Standard packages are not included in the table.

Package name	Link
h5py	https://www.h5py.org (2)
Pandas	https://pandas.pydata.org (3)
Scipy	https://www.scipy.org (4)
Matplotlib	https://matplotlib.org (5)
Lumicks.pylake	https://lumicks-pylake.readthedocs.io

SUPPORTING REFERENCES

1. Harris, C. R., K. J. Millman, . . . T. E. Oliphant. Array programming with NumPy. Nature 2020 585(7825):357-362, doi: 10.1038/s41586-020-2649-2.
2. Collette, A. Python and HDF5. O'Reilly; 2013
3. McKinney, W. Data Structures for Statistical Computing in Python. Proc. of the 9th Python in Science Conf. 2010; 445:51-56. doi: 10.25080/Majora-92bf1922-00a
4. Virtanen, P., R. Gommers, . . . Y. Vázquez-Baeza. 2020. SciPy 1.0: fundamental algorithms for scientific computing in python. Nat. Methods 2020; 17(3):261-272, doi: 10.1038/s41592-019-0686-2.
5. Hunter, J. D. 2007. Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering. 9(3):90-95, doi: 10.1109/MCSE.2007.55.