Data analysis methods for solid-state nanopores

Calin Plesa, Cees Dekker*.

Department of Bionanoscience, Kavli Institute of Nanoscience, Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands

* Corresponding author

Abstract

We describe a number of techniques for the analysis of solid-state nanopore ionic current traces and introduce a new package of Matlab analysis scripts with GUI frontends. We discuss methods for the detection of the local baseline and propose a new detection algorithm which bypasses some of the classical weaknesses of moving-average detection. Our new approach removes detected events and re-creates an ideal event-free baseline which is subsequently used to recalculate the local baseline. Iterative operation of this algorithm causes both the moving average of the baseline current and its standard deviation to converge to their correct values. We explain different approaches to selecting events and building event populations, and we show the value of keeping track of the changes in parameters such as the event rate and the pore resistance throughout the course of the experiment. Finally, we introduce a new technique for separating unfolded events and detecting current spikes present within translocation events. This open source software package is available online at: http://ceesdekkerlab.tudelft.nl/downloads/

Keywords: nanopore, software, analysis, algorithm

1. Introduction

Over the past decade, there has been tremendous growth and progress in research on solid-state nanopores[3, 15]. In this technique, a membrane containing a nanometer-scale pore is placed in-between two chambers containing an electrolyte solution, as shown in Figure 1a. An electric field is applied across the membrane and charged molecules, such as DNA, present in the solution experience an electrophoretic force which pulls them towards the pore and causes them to translocate through. As a molecule translocates through the nanopore, it temporarily blocks the current and this causes a temporary resistive pulse, as shown in Figure 1b. Typically, the duration of the pulse contains information about the length of the molecule while its amplitude is dependent on the molecule's cross-sectional volume.

With the steep development of this field has come the need for signal-processing tools specifically suited to this niche. While many different techniques exist for analysis of nanopore current traces, the majority of data analysis is done on custom software which differs from lab to lab, although some approaches have recently been published[1, 6, 10]. Arjmandi et al[1] have discussed the advantages of wavelets over low-pass filtering, particularly in the accurate recovery of the dwell time and amplitude of translocation events. Raillon et al[10] have proposed a new level-fitting algorithm based on the cumulative-sums algorithm. Pedone et al[6] focused on the accurate analysis of short pulses, which is a common issue in experiments aimed at detecting proteins and short DNA.

In this paper, we describe the many aspects of nanopore data analysis as combined in one single comprehensive new Matlab GUI-based package named Transalyzer. We also introduce novel approaches for detecting the local baseline, extracting current peaks present within events, and we describe various analysis strategies for specific scenarios. Our analysis procedure is split into three successive stages, with each stage utilizing parameters determined in the previous stage, as shown in Figure 1c. The first stage (GUI_detect) determines the local baseline, rms noise level (σ), it detects each translocation event, and determines its basic

properties such as duration, current blockade level, and integrated area [event charge deficit (ECD)]. In the second stage (GUI_events), mixed event populations are sorted and population level statistics are generated, such as the most probable dwell time, blockade level, and event rate. The third and final stage (GUI_localstructures) reanalyzes each event in a given population for the presence of local structures such as bound protein or knots. This analysis pipeline allows us to address the large variability encountered in different types of experiments.

2. Event detection and characterization

The analysis procedure begins with the detection of translocation events within a noisy baseline. As most other labs, we use a thresholding algorithm to extract events. In this approach, events are identified if they cross a threshold (typically 5σ) away from the local baseline level. The threshold is defined by multiplying a *peak detection factor* and the rms noise level (σ), as shown in Figure 2. The *peak detection factor* is chosen large enough to minimize the number of noise spikes captured, while simultaneously low enough to capture as many translocation events as possible. Successful detection of translocation events requires proper identification of the local value of the baseline and the noise level (σ). A variety of factors complicate the determination of these two values, including: (1) inherently unstable baselines, (2) very large event rates, (3) pore clogging, and (4) successive closely-spaced events. Here, we describe how we have addressed some of these issues in our analysis software, which has been used to analyze a large variety of experimental data.

2.1 Baseline detection

Traditionally, baseline detection is performed by calculating a moving average, with the window size optimized to the maximum period of time over which the baseline value is allowed to fluctuate by some chosen amount. Issues affecting proper baseline detection can be distinguished based on the baseline's stability. In the case of a stable baseline, the size of the window can be kept quite large (say, 30k points at 500k samples/s) and this can provide an

accurate value in most cases. In cases where the baseline is unstable, however, the window size must be kept small (<6k points) in order to track the baseline fluctuations. In both of these cases, particularly the latter, the moving average can become an inaccurate representation of the local baseline due to the fact that previous events influence the value of the local baseline. This effect is especially noticeable if the event rate is very high, leading to many closely spaced events, or if the event durations are a significant fraction of the window size. We introduce a simple algorithm to deal with all these issues in Section 2.3.

2.2 Noise level determination

A number of techniques exist for determining the noise level. In the case of a stable baseline, with short-duration events at a low event rate, the trace file can be split into small segments and the global standard deviation can be used. A more accurate method, which works well with high-event-rate data sets, is to first determine the standard deviation in a small moving window (typically 1000 points in size). The values of the standard deviation for all of the windows in a trace segment can be put into a histogram where the bin width is defined by the precision required. The center of the main peak in the resulting histogram typically provides an accurate value of the standard deviation within the trace.

2.3 Iterative detection algorithm

In order to overcome the limitations of the thresholding approach, we designed and implemented a new algorithm shown in Figure 3a. This approach involves iterating through the thresholding algorithm multiples times in order to decouple the moving average calculation of the local baseline from the influence of previous events. At the end of each iteration a new current trace is generated where the duration of each detected event is replaced by the value of the local baseline at the start of the event. An assumption is thus made that the baseline value does not change significantly (>σ) over an average event's translocation time, which holds true in the majority of the experimental data we have encountered. This new trace is subsequently used to recalculate the moving average (using the same approach described in

section 2.1), the rms noise level, and detect the events again. In Figure 3b we show a comparison of the value of the local baseline for a 20k moving average, a 5k moving average, and a 5k moving average with 2 iterations of the algorithm. In this simulation, three events are placed close together. Using the small 5k-point moving average results in a very inaccurate value of the local baseline for the second and third events. Increasing the window size to 20k points improves the accuracy but fails to completely eliminate this effect. The proposed algorithm quickly converges to the correct value with each iteration, while allowing small window sizes to be used, as shown by the 5k-point 2-iteration trace. Similarly for the calculation of the standard deviation, a second trace is created after each iteration were the events are removed and this is subsequently used to determine the new value of the standard deviation. The iterative algorithm is capable of handling event rates where the average time between events is twice as small as the size of the moving window used. So a 5k point window, corresponding to 10 ms at 500k Samples/s, can be used on data with event rates of 200 Hz, as long as the average translocation time of the events is several times smaller than the size of the moving window. Much higher event rates must be addressed on a case-by-case basis, although these situations are typically avoided because they can lead to multiple molecules within the pore simultaneously, which can significantly complicate analysis. In the future several alternative implementations of the iterative algorithm could be used to handle more unstable baselines, at the cost of increased computational time. This could include using both the forward and backward moving averages to determine starting and ending points for the event, and interpolating the change in the baseline that occurred over the course of the event.

How does the iterative algorithm perform when analyzing experimental data? We can quantify the improvement in the value determined for the local baseline by introducing a new measure $\langle I_{\Delta B} \rangle$. This is calculated by first finding the mean value of the fifty points preceding the start of this event and subsequently determining the difference between this mean and the value of the local baseline (from the moving average) for each particular event. We take the absolute value of this difference and determine the mean ($\langle I_{\Delta B} \rangle$) and standard deviation (STD) of the

resulting distribution. If the value of the baseline improves, we expect the value of $<I_{\Delta B}>$ to reduce and the spread of its distribution to become more narrow. We applied this approach to several DNA and protein experimental datasets and reanalyzed each dataset using 0, 1, or 2 iterations of the algorithm, with the results shown in Table 1. We observed reductions in $<I_{\Delta B}>$ and STD after one iteration in all cases, with further iterations bringing minimal improvements. The larger improvements observed in DNA experiments, can be attributed to the longer duration of these events compared to proteins, which leads to larger changes in the moving average. Although the changes may appear small, these values are averaged over thousands of events. This simple algorithm can thus improve the analysis results and overcome the issues associated with thresholding detection.

2.4 Event characterization

Proper determination of each event's characteristics (duration, blockade, and ECD) can be complicated by many types of physical phenomena and data-handling effects, depending on the type of experiment, including: short events prone to filtering distortions[6], low SNR, long tail events, folding[8, 14], events were the current increases rather than decreases during translocation[13], hybrid events were the current both decreases and increases[5], events where the molecule docks onto the pore before translocating[2, 7, 12, 16], knotting[11], mixed populations, pore growth over time[4], biomolecule-pore interactions[9], protein-DNA interactions, and the presence of short DNA fragments. For the event duration, using the full-width-half-maximum value (in conjunction with a Gaussian low-pass filter), provides the most accurate translocation time value, even in the light of various distortions introduced by filtering[1, 6]. The blockade level for very short duration events (short DNA or proteins) is best represented by the maximum blockade value. For longer events, dividing the ECD by the FWHM time provides the best representation of the blockade level for many different types of events. For blockades with well-defined levels such as large folds[8], level-fitting software such as OpenNanopore[10] can be used. We have added an export function into Transalyzer capable of exporting event databases into OpenNanopore, effectively acting as an event pre-processor.

Our software allows the user to select between multiple analysis techniques to determine the translocation time and blockade level of each population, since multiple populations can coexist within the same experiment.

Due to the many different types of event blockades possible, we allow the user to select between three different types: current increase, current decrease, and hybrid (decrease and increase). This feature can also be used in situations where there is a very low SNR by exploiting the fact that noise is symmetric around the baseline while translocation events (typically) are not. In our approach, the same dataset is analyzed twice, once assuming current increase and again using current decrease. Differences in the properties of the resulting populations, such as the event rate, provide strong evidence that translocation events are present, even when it is difficult to differentiate individual events from noise.

3. Population sorting and characterization

Events can be sorted using a number of different criteria into different populations. Our software allows the user to set a minimum and maximum translocation time, current blockade, local baseline level, event number, and event charge deficit in order to select out an event population. In most situations, the event charge deficit (ECD) has a Gaussian distribution for a population of molecules with homogeneous length. A non-Gaussian or distorted ECD distribution can be caused by significant molecule fragmentation, strong biomolecule-pore interactions, low SNR, the presence of docking levels, or overlapping populations. Importantly, selecting a population using the ECD allows folded events to be included in the selection. If folding is not possible, due to the nature of the analyte (nanoparticle, globular protein, etc..) or because the size of the pore is too small, selection using the translocation time can also be useful. Unfolded events (*i.e.* events with no extra peaks present) can be selected by looking at the maximum amplitude distribution, where similar to current histograms, events contribute to Gaussian peaks depending on the folding, with the first peak corresponding to unfolded events. Selection on event number can be used in time-dependent processes where conditions change during the

experiment. Finally, selecting using each event's local baseline allows the quick removal of clogs as well as, if preferred, translocation events which occurred while the pore was partially blocked. Once an event population is selected, it can be characterized using well-established properties such as the most probable translocation time, the most probable blockade amplitude, the most probable ECD, and the event rate.

Keeping track of how properties change over time during an experiment can be quite useful in many instances. Fluctuations in the event rate as a function of time can indicate the presence of a number of processes: Sudden changes in the event rate can indicate the presence of a clog or partial pore blockage. A slow decrease in the event rate over time suggests possible adsorption of the analyte to the pore membrane or flowcell walls, as can, for example, occur with DNA sticking to SiN in the presence of divalent cations. A gradual increase in the event rate at the start of the experiment which subsequently reaches a plateau level is indicative of poor mixing conditions in the flowcell, an effect noticeable with high-viscosity buffers. Tracking of the absolute value of the baseline as a function of time can be used to quantify effects such as pore growth. Indeed, for long-duration experiments where the baseline value is observed to change significantly over the course of the experiment, the amplitude of the events should be normalized by the value of the local baseline in order to be comparable to each other. This issue is particularly relevant in measurements on small diameter pores. These issues highlight the benefit of tracking how global properties change over the course of an experiment.

4. Local structures detection

Finally, we briefly describe how the presence of small current spikes within a translocation event can be detected. Such analysis can, for example, be useful for experiments involving DNA-bound proteins or DNA knots. We begin by separating events containing large folds from unfolded events that contain local spikes. This is accomplished by looking at the area occupied by the current trace in between the first two DNA blockade levels (I_1 and I_2). The first blockade

level (I_1) is the most probable blockade level with only a single (double-stranded) DNA molecule inside the pore, while the second blockade level (I_2) is the most probable blockade level when two DNA molecules are in the nanopore simultaneously. These two levels can be determined from their respective peaks in a current histogram. Figure 4ab provides two example events, one unfolded and one with a large fold at the start. The area occupied by the current trace (between I_1 and I_2) is shown in red in Figure 4cd, while the product ($I_1 t_{FWHM}$) of the DNA blockade level (I_1) and the FWHM translocation time of the event (t_{FWHM}) is represented by a green rectangle. The area occupied by the current (red) is normalized using this value (green) to produce the normalized charge deficit between I_1 and I_2 ($NCD_{1\text{-}2}$). Events with large folds have a large value of $NCD_{1\text{-}2}$ while unfolded events with spikes have smaller values. For example, the event in Figure 4a has $NCD_{1\text{-}2}$ = 0.125 while the folded example of Figure 4b has $NCD_{1\text{-}2}$ = 0.350. Circular molecules produce $NCD_{1\text{-}2}$ values close to 1. Figure 4e shows a typical distribution of $NCD_{1\text{-}2}$ values for an experiment of DNA with bound proteins that translocate through a 20 nm pore. In order to determine a cutoff between folded and unfolded events, we look at known folding rates from DNA-only experiments. For example, in 1M KCl at 30 kHz bandwidth in a 20 nm pore (i.e. the same conditions of the experiment of Fig. 4e), lambda-phage DNA is observed to have approximately 36% of events unfolded. Figure 4f shows the normalized cumulative sum of the $NCD_{1\text{-}2}$ distribution. A horizontal blue line has been added at a value of 0.36; a vertical blue line defined by the point of intersection (in this case at $NCD_{1\text{-}2}$ = 0.22) between the curve and 0.36 provides the cutoff value used to define events as unfolded or folded. Once a dataset is generated with only unfolded events, we then detect peaks present within the DNA event. Essentially our analysis comes down to detecting events within events. For each peak detected we record the temporal position, the position normalized with the total event duration, the peak FWHM, and the peak amplitude. This simple approach allows for the quick separation of folded and unfolded events and the subsequent detection of any local structures present.

4. Discussion and conclusions

We have described a number of analysis techniques implemented in our analysis software and provided a number of examples for specific scenarios. Unlike previous works, we have addressed various effects which occur throughout the analysis procedure. The iterative detection algorithm that we have described provides a simple way to overcome issues typically encountered when using the thresholding detection approach. Furthermore, we have outlined a new method for separating folded events from unfolded events containing current spikes, which is particularly useful in the detection of local structures.

Our Transalyzer analysis package has been licensed under the New BSD Licence, which encourages further development and modification by other labs by imposing minimal restrictions on its modification and redistribution. It is freely available for download from our lab website (http://ceesdekkerlab.tudelft.nl/downloads/). A Subversion repository has also been created on Google Code (http://code.google.com/p/transalyzer/) to encourage future improvements, additions, and code modifications by other labs.

Acknowledgments

We thank Auke Booij and Mathijs Rozemuller for code contributions. We also thank Sanne de Jongh, Ruben Zinsmeester, Bruno van den Toorn, Stephanie Heerema, Magnus Jonsson, Adithya Ananth, Daniel Verschueren, Francesca Nicoli, and Claire Hurkmans for testing, reporting bugs, and suggesting new features. This work was supported in part by the European Research Council Advanced Grant NanoforBio (no. 247072) as well as the Netherlands Organisation for Scientific Research (NWO/OCW), as part of the Frontiers of Nanoscience program.

Figures

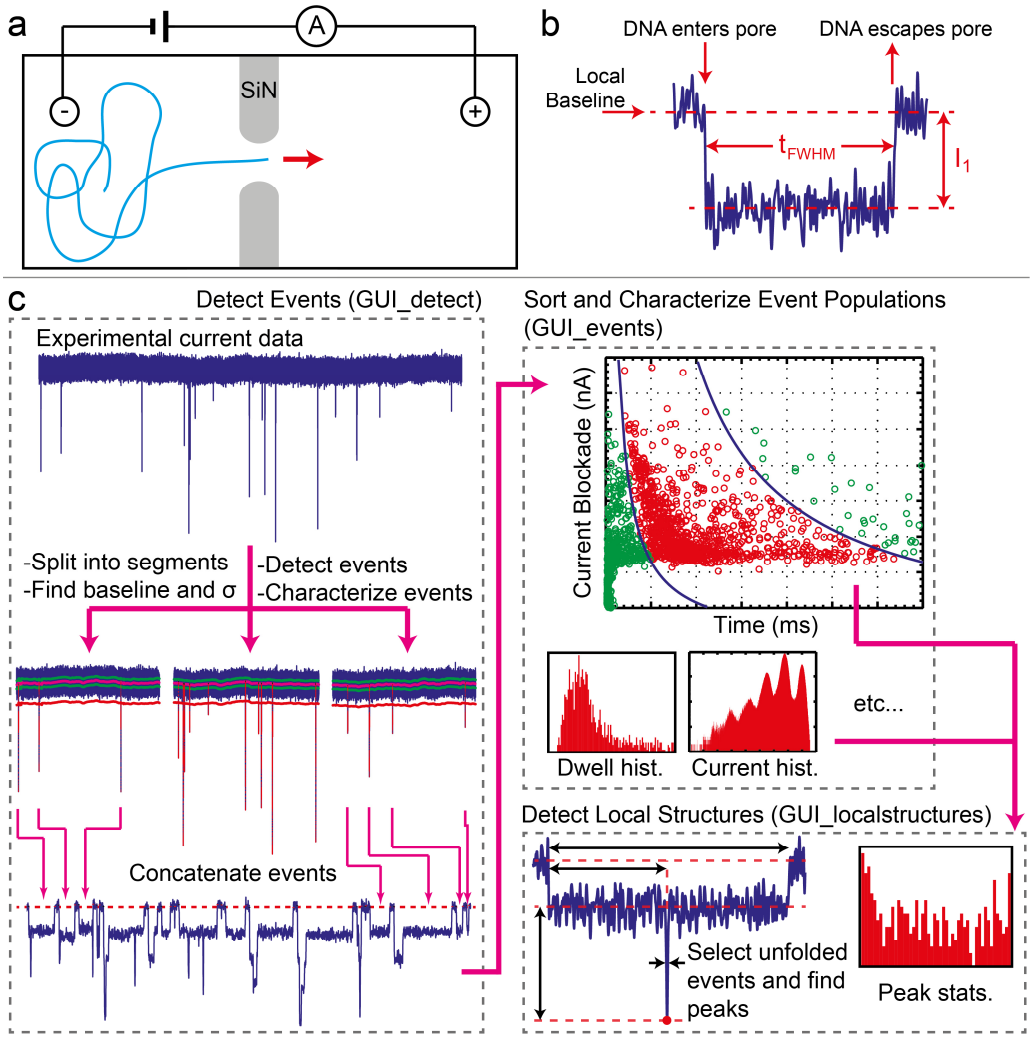

Figure 1. a) Illustration of a typical nanopore setup. **b)** Current signal produced by a translocating DNA molecule. **c)** Schematic of the typical analysis procedure of a nanopore current trace, which is divided into three parts. The first part splits a current file, detects the events in each segment, characterizes each event, and concatenates all found events. The second part sorts and characterizes the event populations. The final part can be used to sort and re-analyze events for the presence of local structures and to generate relevant statistics.

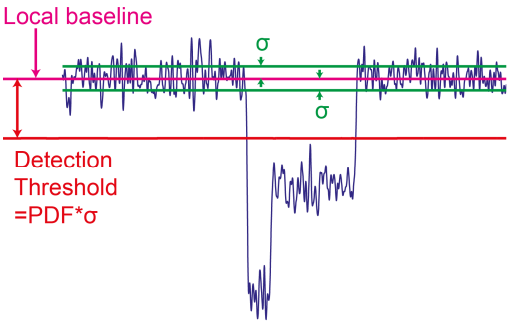

Figure 2. A typical threshold detection scheme involves finding the local baseline and rms noise level (σ). A detection threshold is set as a constant (PDF; peak detection factor) multiplied by the rms noise level, away from the local baseline. Events are detected by finding points where the current trace crosses the detection level. The trace can be analyzed backward from the first crossing point and forward from the next crossing point to find the points where the current crosses the local baseline, which define the start and stop of the event.

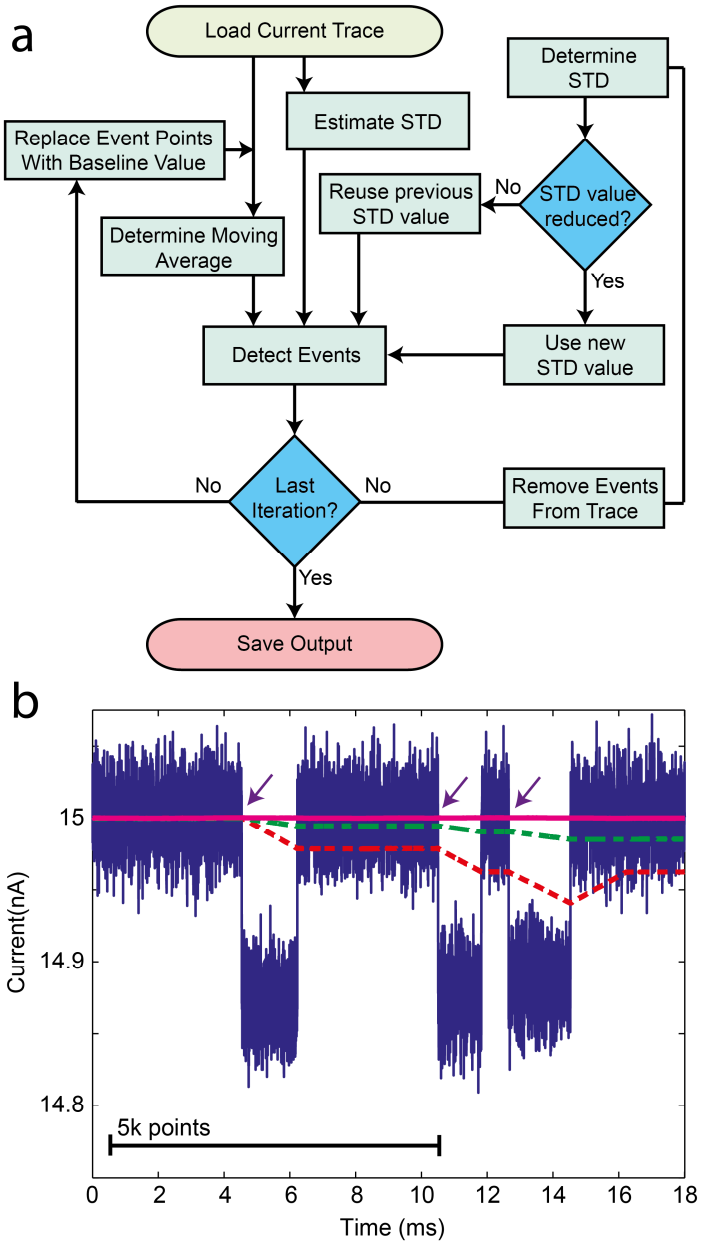

Figure 3. a) Flowchart of the iterative detection algorithm. After each iteration, information about the events found feeds back into the next iteration to improve the value of the local baseline and rms noise level. **b)** A simulation of three closely spaced translocation events, with the local baseline determined using three different techniques. Moving averages of 20k and 5k points, represented by the dashed green and red lines, fail to properly determine the local baseline of the 2nd and 3rd events because the moving average is influenced by the previous events. The solid magenta line shows the same 5k-point moving average after 2 iterations of

the detection algorithm, demonstrating that it is able to accurately determine the value of the local baseline despite using a small window size.

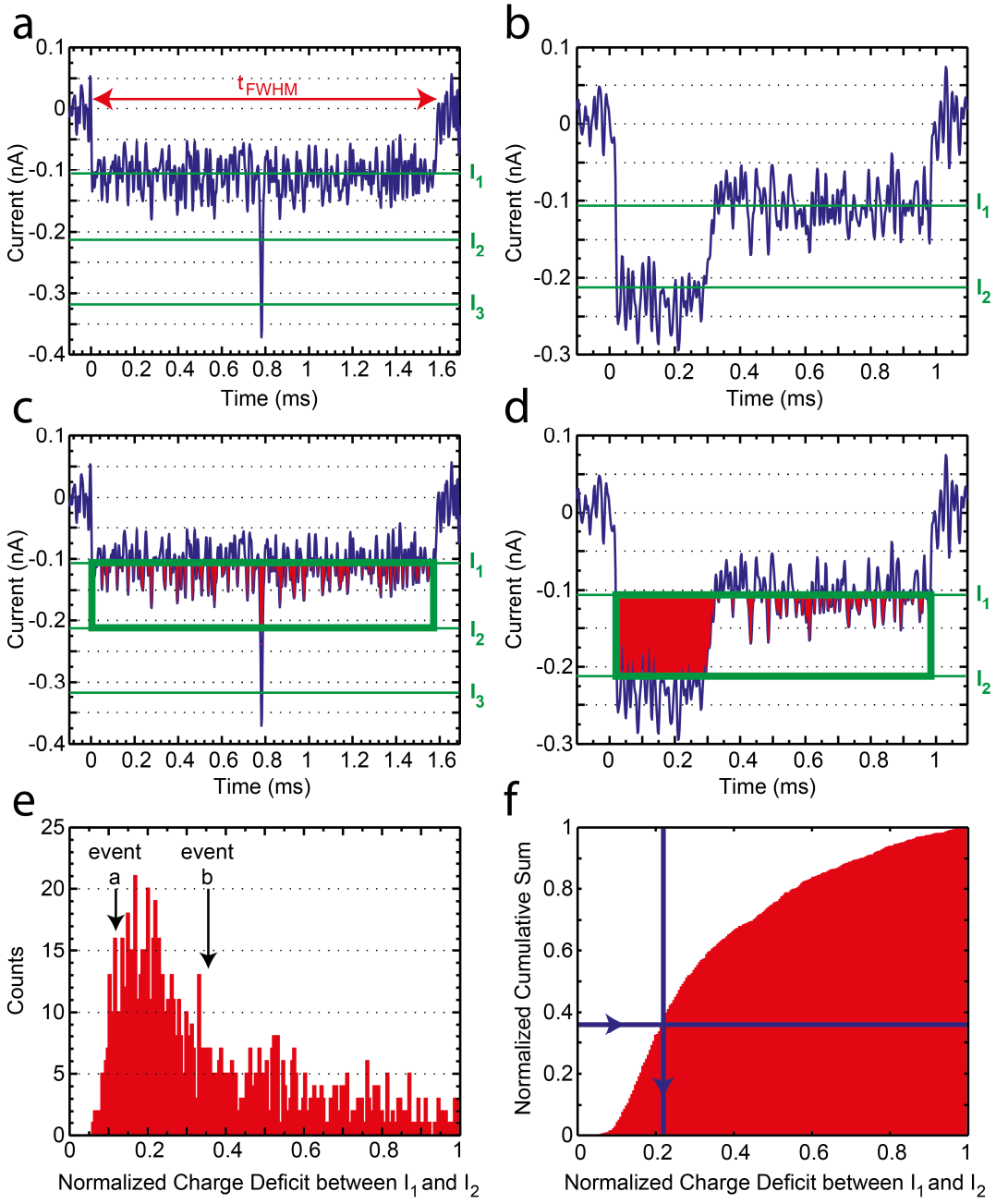

Figure 4. Detection of local peaks within events. **a)** Current trace of an unfolded event with a single spike. **b)** Current trace with a folded event. The horizontal green lines represent multiples of the single dsDNA blockade level (I_1) as determined using a current histogram. **c-d)** The same events as in a-b with the area in between the first two blockade levels highlighted. The integral (charge deficit) of the current trace between these two levels is shown shaded in

red. This charge deficit is normalized by the total area given by the product ($I_1 t_{FWHM}$) of the DNA blockade level I_1 and the FWHM translocation time t_{FWHM} of the event, shown as a green rectangle. The resulting value is termed the normalized charge deficit between I_1 and I_2 (NCD_{1-2}). Events with folds have higher NCD_{1-2} values. **e)** Typical distribution of NCD_{1-2} values for a protein-DNA experiment where DNA events contain short spikes, along with the positions of the two example events of panel a and b. **f)** Normalized cumulative histogram for the distribution shown in e. The vertical line shows the proportion of events which are typically unfolded in these conditions as determined using DNA-only control experiments. The vertical line is the intercept of the normalized cumulative sum with this, and is used to determine the maximum NCD_{1-2} value allowed for an event to be considered unfolded.

Table 1 – In order to quantify the improvement in the calculation of the baseline we determine $I_{\Delta B}$, which is the absolute value of the difference between the mean of the fifty points preceding the start of the event and its local baseline value. The mean ($<I_{\Delta B}>$) and standard deviation of $I_{\Delta B}$ values in each dataset is shown. All experiments were carried out in 1M KCl, filtered at 10 kHz, and analyzed with a 5k point moving average. Improvements in the value determined for the baseline results in lower values of $<I_{\Delta B}>$ and its STD. In all cases we see an improvement after one iteration, with further iterations bringing only minimal improvements.

Dataset		*Num. of Events*	*Event rate (Hz)*	*Num. of Iter.*	$<I_{\Delta B}>$ *(pA)*	*STD (pA)*
A	*λ DNA, 20 nm pore, 100 mV*	1975	4.8	0	6.3	6.2
				1	5.5	4.7
				2	5.4	4.4
				3	5.4	4.4
B	λ DNA, 10 nm pore, 500 mV	1477	10.0	0	23.4	23.7
				1	20.9	20.1
				2	20.9	20.1
C	T4 DNA, 20 nm pore, 100 mV	1287	0.3	0	4.7	4.1
				1	4.6	3.7
				2	4.6	3.7
D	IgG antibody, 20 nm pore, 100 mV	10221	162.2	0	6.6	6.7
				1	6.0	6.0
				2	5.9	6.0
E	99kDa protein, 16 nm pore, 100 mV	6009	35.8	0	7.0	6.2
				1	6.9	6.1
				2	6.9	6.1

References

[1] Arjmandi N, Roy W V, Lagae L and Borghs G 2012 Improved Algorithms for Nanopore Signal Processing *arXiv*

[2] Carlsen A T, Zahid O K, Ruzicka J, Taylor E W and Hall A R 2014 Interpreting the Conductance Blockades of DNA Translocations through Solid-State Nanopores *ACS Nano* **8** 4754-60

[3] Dekker C 2007 Solid-state nanopores *Nat Nano* **2** 209-15

[4] Hout M v d, Hall A R, Wu M Y, Zandbergen H W, Dekker C and Dekker N H 2010 Controlling nanopore size, shape and stability *Nanotechnology* **21** 115304

[5] Kowalczyk S W and Dekker C 2012 Measurement of the Docking Time of a DNA Molecule onto a Solid-State Nanopore *Nano Lett.* **12** 4159-63

[6] Pedone D, Firnkes M and Rant U 2009 Data analysis of translocation events in nanopore experiments. *Anal. Chem.* **81** 9689-94

[7] Plesa C, Ananth A N, Linko V, Gülcher C, Katan A J, Dietz H and Dekker C 2013 Ionic Permeability and Mechanical Properties of DNA Origami Nanoplates on Solid-State Nanopores *ACS Nano* **8** 35-43

[8] Plesa C, Cornelissen L, Tuijtel M W and Dekker C 2013 Non-equilibrium folding of individual DNA molecules recaptured up to 1000 times in a solid state nanopore *Nanotechnology* **24** 475101

[9] Plesa C, Kowalczyk S W, Zinsmeester R, Grosberg A Y, Rabin Y and Dekker C 2013 Fast Translocation of Proteins through Solid State Nanopores *Nano Lett.* **13** 658-63

[10] Raillon C, Granjon P, Graf M, Steinbock L J and Radenovic A 2012 Fast and automatic processing of multi-level events in nanopore translocation experiments *Nanoscale* **4** 4916-24

[11] Rosa A, Di Ventra M and Micheletti C 2012 Topological Jamming of Spontaneously Knotted Polyelectrolyte Chains Driven Through a Nanopore *Phys. Rev. Lett.* **109** 118301

[12] Rosenstein J K, Wanunu M, Merchant C A, Drndic M and Shepard K L 2012 Integrated nanopore sensing platform with sub-microsecond temporal resolution *Nat Meth* **9** 487-92

[13] Smeets R M M, Keyser U F, Krapf D, Wu M-Y, Dekker N H and Dekker C 2006 Salt Dependence of Ion Transport and DNA Translocation through Solid-State Nanopores *Nano Lett.* **6** 89-95

[14] Storm A J, Chen J H, Zandbergen H W and Dekker C 2005 Translocation of double-strand DNA through a silicon oxide nanopore *Phys. Rev. E: Stat., Nonlinear, Soft Matter Phys.* **71** 051903

[15] Wanunu M 2012 Nanopores: A journey towards DNA sequencing *Physics of Life Reviews* **9** 125-58

[16] Wei R, Martin T G, Rant U and Dietz H 2012 DNA Origami Gatekeepers for Solid-State Nanopores *Angew. Chem., Int. Ed.* **51** 4864-7