Fast and automatic processing of multi-level events in nanopore translocation experiments

Camille Raillon, Pierre Granjon, Michael Graf, Lorentz Steinbock, Aleksandra Radenovic

Nanoscale

Cite this: DOI: 10.1039/c2nr30951c

www.rsc.org/nanoscale

FEATURE ARTICLE

Fast and automatic processing of multi-level events in nanopore translocation experiments†

C. Raillon,[a] P. Granjon,[b] M. Graf,[a] L. J. Steinbock[a] and A. Radenovic*[a]

Received 20th April 2012, Accepted 13th June 2012
DOI: 10.1039/c2nr30951c

We have developed a method to analyze in detail, translocation events providing a novel and flexible tool for data analysis of nanopore experiments. Our program, called OpenNanopore, is based on the cumulative sums algorithm (CUSUM algorithm). This algorithm is an abrupt change detection algorithm that provides fitting of current blockages, allowing the user to easily identify the different levels in each event. Our method detects events using adaptive thresholds that adapt to low-frequency variations in the baseline. After event identification, our method uses the CUSUM algorithm to fit the levels inside every event and automatically extracts their time and amplitude information. This facilitates the statistical analysis of an event population with a given number of levels. The obtained information improves the interpretation of interactions between the molecule and nanopore. Since our program does not require any prior information about the analyzed molecules, novel molecule–nanopore interactions can be characterized. In addition our program is very fast and stable. With the progress in fabrication and control of the translocation speed, in the near future, our program could be useful in identification of the different bases of DNA.

1. Introduction

Nanopores are nanometric holes in thin insulating membranes existing in two modalities, protein/biological pores and solid-state pores. Protein pores are made using a pore-forming protein such as α-hemolysin[1] that is inserted in a lipid bilayer whereas solid-state pores are fabricated in an insulating membrane using highly focused ions[2] or electrons.[3]

[a]*Laboratory of Nanoscale Biology, Institute of Bioengineering, School of Engineering, EPFL, 1015 Lausanne, Switzerland. E-mail: aleksandra.radenovic@epfl.ch*
[b]*Grenoble Image Speech Signal Automatics Laboratory, Grenoble Institute of Technology, Grenoble, France*
† Electronic supplementary information (ESI) available. See DOI: 10.1039/c2nr30951c

C. Raillon

Camille Raillon received her M.Sc. in Electrical Engineering with Honors in 2007 from Grenoble Institute of Technology (INPG). During her master's, she participated in a student exchange program at the University of Sherbrooke, Canada where she did her master's thesis: fast prototyping of microfluidic channels in SU-8 photoresist for BioMEMS. After graduating from INPG, Camille worked as a characterization engineer for a year in a MEMS-based pressure sensors company in Silicon Valley, California. Camille is now a Ph.D. student at EPFL. Her Ph.D. project focuses on the single-molecule study of transcription using nanopore sensing.

P. Granjon

Pierre Granjon received his Ph.D. degree from the Grenoble Institute of Technology (INPG), France in 2000. He joined the Laboratoire des Images et des Signaux (LIS) in 2002 and the Gipsa-lab at INPG in 2007, where he currently holds a position as associate professor. His current research area mainly concerns signal processing for condition monitoring and power system monitoring (power systems, power networks, batteries…). He is particularly interested in linear and non-linear optimal filtering, multi-component signal analysis and sequential change detection algorithms.

Nanopores are used as biosensors for single-molecule detection; they can detect unlabeled biopolymers such as DNA and RNA,[4–8] single proteins,[9,10] ligand or protein–DNA complexes[11–13] and also RNA–antibiotic complexes.[14]The detection method is simple: when a molecule passes through the nanopore the ionic current is significantly reduced because the regular flow of ions passing through the pore is blocked. While nanopore detection of those molecules has been extensively studied and optimized, data analysis is still not standardized and can be very challenging. As a preferred analyzing tool, most of the groups use time distribution of ionic current to classify the hundreds of events that are collected in a single experiment. Such a histogram (called a point histogram here in the text) can be used to identify peaks in the current signal.[15] Once those peaks have been identified, event extraction can be done with a computer-based program using a threshold. Finally, the mean blockage and the event duration (or dwell time) can be calculated. This method performs well and is commonly used but lacks information on the different levels inside each event.

Using this method on data displaying low noise, Meller *et al.*[7] were able to discriminate between single polynucleotide molecules using scatter plots, dwell time and current blockade point histograms. Other studies[16,17] successfully identified populations amongst DNA translocations through small nanopores (2–5 nm in diameter) and linked those populations with molecule–nanopore interactions. Dwell time histograms have also been used to perform molecule sorting,[6] where λ DNA and fragments of λ DNA digested by the restriction enzyme HindIII can be differentiated. This work relies on the fact that those DNA fragments have different lengths, hence shorter fragments translocate faster than longer fragments. Since the speed at which a DNA molecule translocates the pore varies significantly over an experiment,[18] finer analysis is required and it is typically performed *via* examination of current blockages that a translocating molecule produces. Here we list examples where the existing method has been successfully applied: Skinner *et al.*[19] for example were able to distinguish between single and double stranded nucleic acids using point histograms of current blockades in solid-state nanopores. Other groups were able to identify nucleotides by immobilizing homopolymers or more complex oligonucleotides in α-hemolysin.[20,21]

As pointed out above, the point histogram technique performs well for current signals with low I_{RMS} but analysis of noisier data still lacks a fast and robust data processing technique. Some commercial software solutions exist, such as pCLAMP from Molecular Devices, Inc. Although pCLAMP is intended for acquisition and analysis of electrophysiology data it can be also used in nanopore data analysis. On the other hand, free software packages such as QuB exist. QuB is based on Hidden Markov Models[22,23] and is intended for the analysis of generalized single-molecule kinetics. Prior knowledge of the signal is required to estimate the statistical model parameters. The group of S. Winters-Hilt has also reported methods to classify and cluster events using hidden Markov models (HMM).[24] Those methods give statistical models of resistivity and dwell times with rate constants and transitions between states. There are other statistical models that have been developed and applied to nanopore data, for example, classification of events using support vector

M. Graf

Michael Graf (born 1989) obtained his B.Sc. in Life Sciences and Technology from EPFL in 2011. He is currently finishing his M.Sc. in bioengineering with specialization in biomedical technologies. His research interests extend over the areas of biophysics, informatics, molecular biology and genomics.

L. J. Steinbock

Lorenz Steinbock studied Molecular Biotechnology (B.Sc. & M.Sc.) at the universities in Heidelberg, Germany, Waterloo, Canada and Cambridge, UK. He graduated in 2006 followed by a Ph.D. in physics at the University of Cambridge, UK in Ulrich Keyser's group. He worked on rupture force experiments with DNA aptamers, pioneered the use of glass nanocapillaries for the detection of DNA using the resistive pulse technique and coauthored the combination with optical tweezers. He obtained prestigious scholarships e.g. from the Deutsche Telekom Stiftung. His current research interest is single-molecule detection with nanopores.

A. Radenovic

Aleksandra Radenovic received her master's degree in physics from the University of Zagreb in 1999 before joining Professor Giovanni Dietler's Laboratory of Physics of Living Matter in 2000 at University of Lausanne. There she earned her Doctor of Sciences degree in 2003. In 2003 she was also awarded a research scholarship for young researchers from the Swiss Foundation for Scientific Research which allowed her to spend 3 years as postdoctoral fellow at the University of California, Berkeley. Before joining EPFL as Assistant Professor in 2008 she spent 6 months at NIH and Janelia Farm. In 2010 she received the ERC starting grant.

machines (SVM)[25] or principal component analysis (PCA).[26] In both cases the information regarding the signal waveform is lost, *i.e.* the shape of the event varying with time.

In this paper we report a novel method for the analysis of signals acquired in nanopore sensing experiments. This new method is fast, automated and requires little prior knowledge of the input signal. So far one group has reported a fitting algorithm to detect levels inside events.[27] This algorithm fits the levels closest to the maxima of a point histogram but the used algorithm is not detailed in the paper and currently not made public. Although the idea of using a change detection algorithm to analyze nanopore data has been presented before[28] it was used to detect events but not to fit levels inside events.

We have chosen to test our method on prototypical translocation data such as the well-studied λ DNA translocation which generate signals that can be easily interpreted and the levels generated have been extensively characterized.

2. Instrumentation and modelling of experimental data

2.1. Experimental setup

The experimental setup is shown in Fig. 1a; this setup is standard for nanopore sensing and is detailed in the ESI.† Our microfluidics has two reservoirs, one on each side of the nanopore. A bias voltage is applied across the nanopore using the electrodes and the ionic current is monitored using an amplifier. Fig. 1b illustrates a typical DNA translocation event. When the negative DNA molecule translocates towards the positive electrode through the nanopore, the base current is significantly reduced because the DNA molecule blocks the regular flow of ions going through the nanopore. The oval 7.2 nm pore used in the two sets of experiments is shown in Fig. 1c. Signals were filtered at a cut-off frequency of 10 kHz and sampled at 100 kHz. The ionic current was amplified and monitored using an Axopatch 200B (Molecular Devices, Inc. Sunnyvale, CA) in resistive feedback mode. Our acquisition system is widely used for nanopore sensing. This makes the acquired data prototypical within the field of nanopore sensing and/or amongst nanopore users.

2.2. Experimental conditions

Prior to DNA translocation experiments, the nanopore was characterized to check for linear current–voltage (*I–V*) characteristics. The DNA was then introduced into the *cis* chamber and a voltage of 100 mV was applied across the nanopore. All the events were recorded using a custom LabVIEW program. This recorded signal was analyzed using our CUSUM method detailed in Section 3.

The results of two experiments are shown in Section 4: the first experiment was recorded at a lower noise level than the second one. Both measurements were done in the same nanopore, the solution was kept at 1 M KCl and the applied voltage at 100 mV. Solid-state nanopores can be reused many times, but after a while the nanopore is more sensitive to the environment and the noise

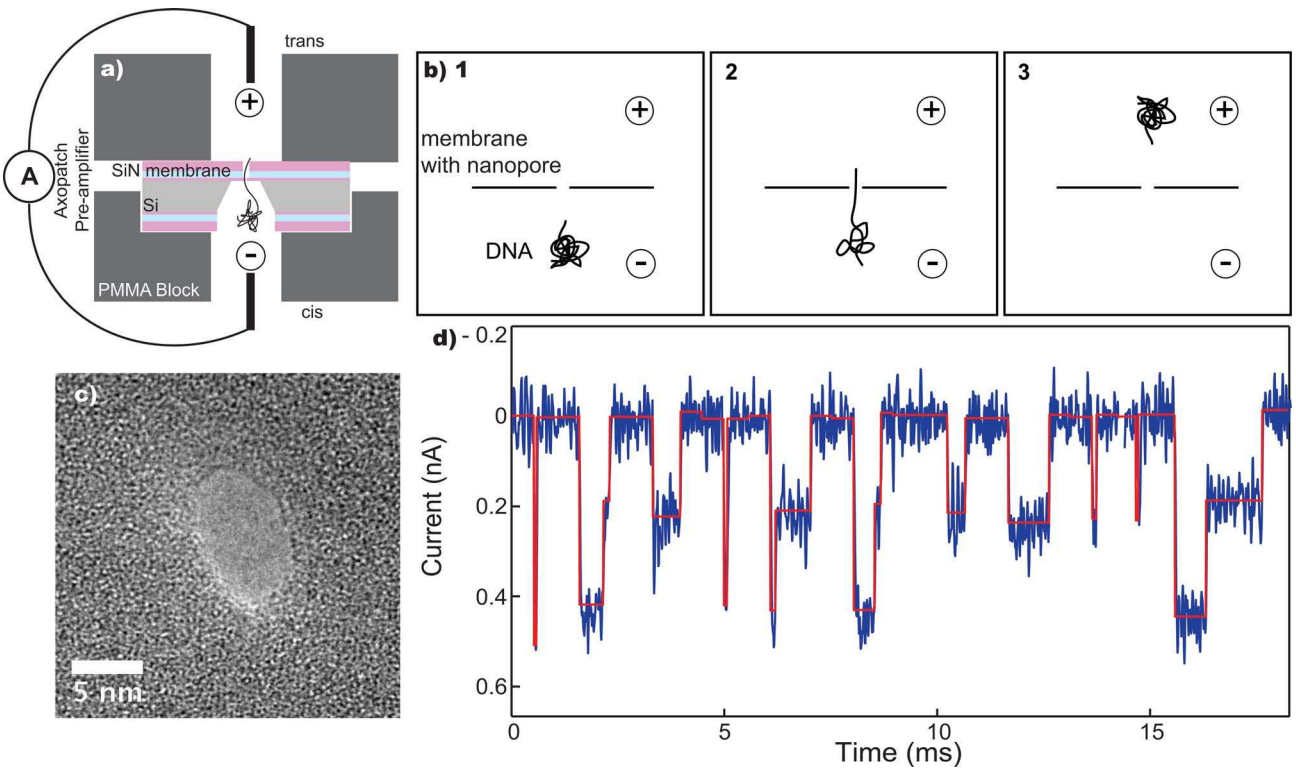

Fig. 1 Schematic of a typical nanopore translocation event and results after CUSUM algorithm. (a) Schematics of the experimental setup. (b) Illustration of a typical DNA translocation event in a nanopore. DNA is negatively charged therefore it translocates through the nanopore towards the positive electrode when a voltage is applied. (c) TEM image of the nanopore used for both measurements. The nanopore was fabricated in a 20 nm thick SiN_x membrane. (d) Concatenation of detected events after rough event detection and segmentation. The raw signal is shown in blue while the fit by the CUSUM algorithm is displayed in red.

level changes. Variation of surface composition can induce surface charge fluctuations and lead to higher 1/*f* noise.[29–31] For both measurements, we evaluated the current standard deviation I_{RMS} and found it to be 36.7 pA_{rms} and 55.3 pA_{rms} for the low and high noise measurements respectively. Those two sets of simple data enabled us to demonstrate the efficiency of our CUSUM method for noisy data and also to compare it with the widely used point histogram method.

2.3. Measurement model (ionic current model)

Our model is based on the premises that the measured signal can be divided into three components using equation (1):

$$i(t) = i_d(t) + \sum_{k=1}^{N} i_{event\_k}(t) + i_n(t) \quad (1)$$

where $i(t)$ is the measured signal, $i_d(t)$ is the base current with drift, $i_{event\_k}(t)$ are the events that are piecewise constant with a given number of levels[5] and $i_n(t)$ is the noise component of the signal.

Let us first discuss the base current $i_d(t)$ which is proportional to the applied voltage because the nanopore is a resistor like component. The conductance of the nanopore is given by a formula that depends on the conductance of the solution and the dimensions of the nanopore.[32] The base current has a quasi-constant value, which drifts slowly compared to the typical duration of an event. Changes in concentration, temperature or a slow modification of the nanopore surface state can induce this low frequency disturbance part.[33–37]

The most important components in our model are the events $i_{event\_k}(t)$. Each event corresponds to the translocation of a DNA molecule in a given configuration. Within the event there could be different levels, for example, when a DNA molecule translocates in a folded conformation.[5] Depending on the size of the translocating molecule, the current can have different piecewise constant values that we call levels. With our CUSUM method we make an automatic fit of those levels, even with a low signal to noise ratio. An event is characterized by a start time and an end time. The start time is defined when a first level is observed away from the base current and the event end time is defined when the signal crosses the base current value again.

Besides baseline and event components, the signal contains noise. For example, the noise component in our two datasets (low and high noise measurements) is displayed in Fig. 2. Fig. 2a shows histograms of both measurements without events, both display a Gaussian distribution. Next, we evaluate the current standard deviation in the frequency domain by taking the square root of the power spectral density (PSD). Both PSDs have been computed using Welch's averaged modified periodogram applied to the data without events (see ESI† for more details). We found the values to be 36.9 pA_{rms} and 55.6 pA_{rms} for the low and high noise measurements respectively. This is in good agreement with the time domain current standard deviation. Fig. 2b shows the superposed PSD plots of the two measurements; the low and high noise measurements are the blue and red curves respectively. The low-pass Bessel filter effect can be identified in both plots. The high frequency noise rising with f^2 also named Johnson noise, originates from the thermal fluctuations of the charge carriers.[33] It is noticeable that this high frequency noise is the same for both low and high noise measurements as it converges around 10 kHz with exactly the same roll-off. The main difference between the low and high noise measurements is clearly the 1/*f* noise by two orders of magnitude. This strong variability of the 1/*f* noise also named flicker noise has already been related to nanopore experiments by Smeets *et al.*[34] and Tabard-Cossa *et al.*[33]

3. The CUSUM method

In the previous section, it has been shown that the measured current can be modeled as a wide-band Gaussian noise added to a piecewise constant signal due to translocation events. In this section, we present a method which can detect such events, and segment the different levels inside these events despite the presence of noise. In the following section, only the sampled version $i(k)$ of this signal will be used.

3.1. General structure

The problem considered in this paper is very similar to the one studied in the statistical process/quality control area, where the condition of a monitored system has to be sequentially

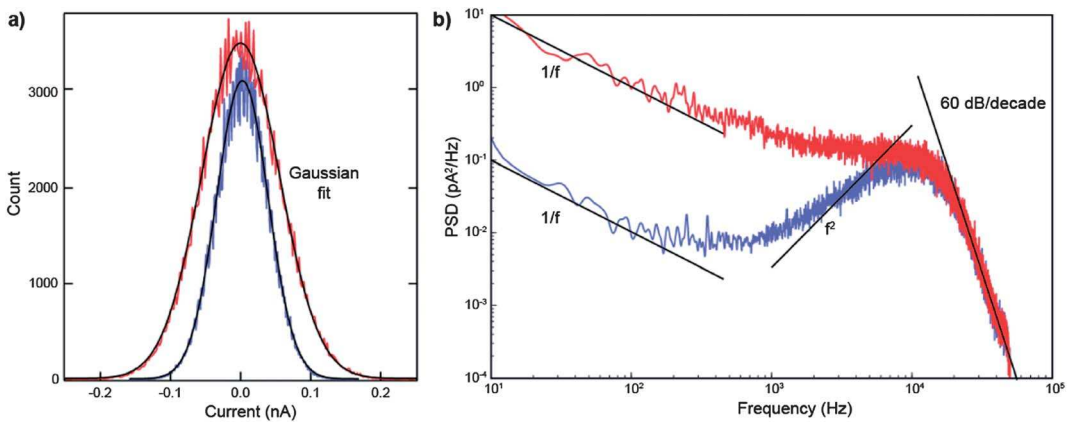

Fig. 2 Histogram and power spectral density of low and high noise measurements. (a) In blue and in red are the histograms of the low and high noise measurements respectively, both showing a Gaussian fit. (b) In blue and in red are the power spectral densities of the low and high noise measurements respectively. The noise is decomposed in two main components, the flicker (1/*f*) and the Johnson f^2 noise.

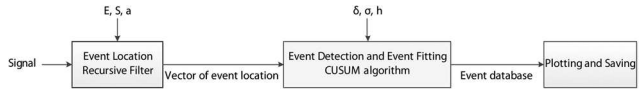

Fig. 3 Flow chart of the OpenNanopore program. The raw signal is processed by the event location subroutine, where a recursive low-pass filter finds rough event locations. A vector with a startpoint and endpoint for each event is then transferred to the CUSUM function. Here, event startpoints and endpoints are detected in a more precise manner and the events are fitted. Next all information from the fitted events is extracted in an event database and finally plotted and saved in the last OpenNanopore subroutine.

determined.[38,39] Indeed, the state of a system is usually monitored through segmentation of noisy piecewise constant fault indicators. Therefore, numerous efficient algorithms have been developed in this community to sequentially detect abrupt changes in signals. A comprehensive survey of this subject is given in Basseville and Nikiforov[40] and Lai.[41]The general structure of the proposed CUSUM method is based upon such sequential algorithms. As shown in Fig. 3 this method mainly consists of three steps:

- the detection of translocation events,
- the segmentation of the detected events into different levels,
- the storage of events and levels information (dwell time, amplitude, *etc.*) in a dedicated database.

These operations are detailed in the following paragraphs.

3.2. Event detection

The goal of the event detection step is to detect and roughly localize translocation events in the measured current. The approach usually applied to detect a translocation event is to apply a threshold. If the baseline has low frequency variations then a simple threshold is not sufficient. One way to avoid such problems is to use adaptive thresholds. In the event detection method detailed in the ESI,† this is realized by defining local thresholds $\eta_S(k)$ and $\eta_E(k)$ through local estimates of the mean $\mu(k)$ and standard deviation $\sigma(k)$ of the current signal. Fig. 4a and b show an example of results obtained with this event detection method and compares the current signal in blue with the local thresholds $\eta_S(k)$ in red and $\eta_E(k)$ in green. It is clear that these two thresholds correctly adapt to the time evolution of the current by following the very low frequency variations. This detection method based on adaptive thresholds follows the low frequency content of the current, and is much less sensitive to the $1/f$ noise than a classical threshold.

3.3. Event segmentation and level fitting

Once translocation events have been detected correctly, the corresponding piecewise constant signals have to be segmented in order to obtain the different levels and the corresponding change times. The first important task consists in identifying impulsive events. Indeed, such events are too short to be composed of different levels and do not have to be further segmented. This identification can be realized by calculating their length due to the start and end samples previously obtained, and by comparing this length with a given number of samples N_{imp}. If the length of one event is less than N_{imp}, it is considered as an impulse and therefore it is not segmented in the next step. On the contrary, if its length is greater than N_{imp}, it may contain different levels and must be segmented. In our case, the maximum length of impulses has been fixed to 0.1 ms, which corresponds to $N_{\text{imp}} = 10$ samples (see ESI† for more details). The impulse length can vary depending on the translocation speed and the sampling frequency.

The second important task is to determine the different levels and change times contained in the events that are not impulsive. The proposed segmentation method relies on a sequential change detection algorithm: the cumulative sums or CUSUM algorithm. The CUSUM algorithm was originally designed for online applications to detect real time changes in production datasets. It is Page[42] who first proposed different forms of this algorithm, direct or recursive, and one-sided or two-sided forms. Later, several authors gave theoretical justifications and foundations of this algorithm.[40,41] Nowadays, this efficient algorithm is widely used in the statistical process/quality control area.[38,39]

This algorithm has been developed under the following assumptions. Let $x(n)$, $n = 0, \ldots, k$ be a discrete random signal with independent and identically distributed samples. Each of them follows a Gaussian probability density function with an expected value μ and a standard deviation σ. This signal may contain one abrupt change occurring at the unknown change time $1 \leq n_c \leq k$. This abrupt change is modeled by an instantaneous modification of μ occurring at the change time n_c. Therefore, $\mu = \mu_0$ before n_c, and $\mu = \mu_0 + \delta$ from n_c to the current sample k, where δ is the change in magnitude to be detected. Under these assumptions, it has been shown by Page[42] and Basseville and Nikiforov[40] that the recursive CUSUM algorithm given in Algorithm 1 is a very efficient sequential algorithm to detect the occurrence of an abrupt change in the signal, and to estimate the corresponding change time n_c.

$k = S(-1) = G(-1) = 0$
while the algorithm is not stopped do
- measure the current sample $x(k)$
- $s(k) = \frac{\delta}{\sigma^2}\left[x(k) - \left(\mu_0 + \frac{\delta}{2}\right)\right]$
- $S(k) = S(k-1) + s(k)$
- $G(k) = \sup\left[0, G(k-1) + s(k)\right]$
  if $G(k) > h > 0$
  - $\hat{n}_c = \underset{1 \leq n \leq k}{\arg\min}\, S(n-1)$
  - stop the algorithm
  end
- $k = k + 1$

end

Algorithm 1**: CUSUM algorithm, recursive form**

In this algorithm, the instantaneous log-likelihood ratio $s(k)$ can be seen as a normalized difference between the current sample $x(k)$ and $\mu_0 + \frac{\delta}{2}$, the arithmetic mean of the expected values before and after the change. The cumulative sum $S(k)$ cumulates these differences, and the decision function $G(k)$

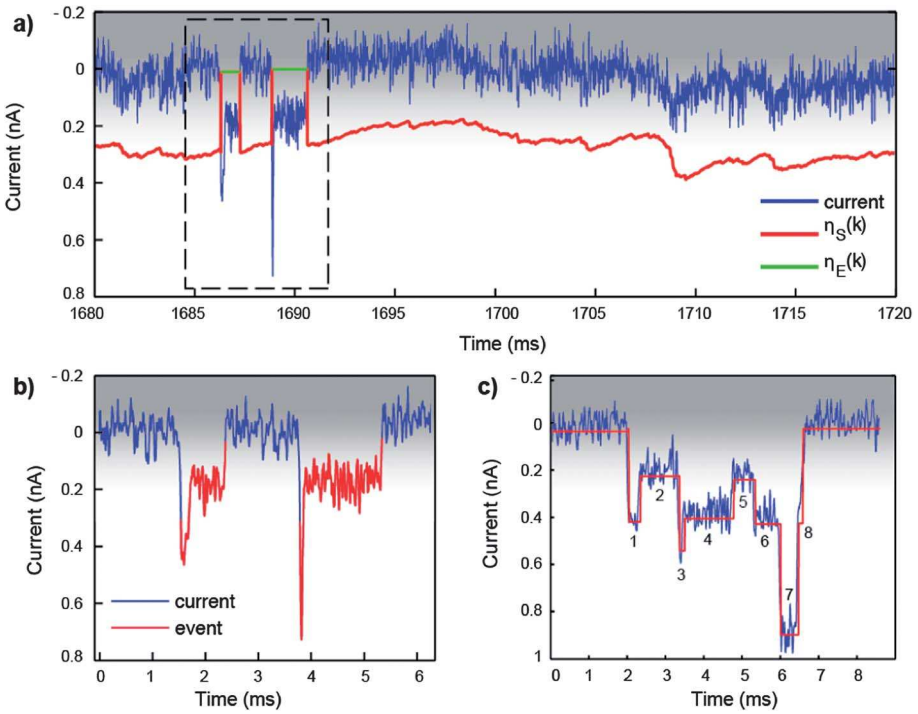

Fig. 4 Event detection done with adaptive thresholding and CUSUM fit of a multi-level event. (a) Results obtained with adaptive thresholds, filter parameters are set to $a = 0.995$, the threshold parameters $S = 5$ and $E = 0$. The low frequency variations around 1709 ms are not identified as events since at the same time, the threshold value $\eta_S(k)$ sufficiently decreases (b) zoom in on the current trace marked with a dashed square displayed in (a) showing that our event detection method correctly detects and localizes the two events occurring around 1609 ms. (c) A multi-level event obtained in λ DNA translocation experiment. One can notice 8 levels in the event that are without difficulty identified and fitted by the CUSUM algorithm.

cumulates their positive part. When the decision function exceeds a user defined positive detection threshold h, the algorithm detects a significant abrupt change somewhere in the past samples. The cumulative sum $S(k)$ is then used to obtain an estimate $\hat{n}_c$ of the unknown change time n_c. The detection threshold h is an important parameter since it is related to the algorithm sensitivity. Indeed, the decision function $G(k)$ needs a large number of samples in order to exceed a high threshold h. In that case, the detection delay is long but the false detection rate is low and the algorithm can be considered as "not sensitive". On the contrary, when h is close to zero, $G(k)$ can exceed this threshold very quickly. The detection delay is then short but the false detection rate increases and the algorithm is "very sensitive". The approach used to correctly set the value of h is given later in this section.

An important characteristic of this algorithm is its optimal performance. Indeed, Lorden[43]and later Moustakides[44] and Ritov[45] demonstrated by different approaches that the CUSUM algorithm is the best sequential algorithm to detect and estimate abrupt changes in discrete random signals with independent and identically distributed Gaussian samples. This property makes this algorithm an excellent fit to analyze translocation events. Indeed, the current model developed in Section 2 with piecewise constant translocation events and added Gaussian noise is very close to these assumptions. However, the CUSUM algorithm presented in Algorithm 1 is difficult to use in practice and had to be adapted. First, the standard deviation σ of the signal and the expected value before change μ_0 have to be known to calculate the instantaneous log-likelihood ratio $s(k)$. This is difficult in our application because these parameters may evolve all along the course of the experiment. Calculating recursive estimates of these two quantities through the past samples easily solves this problem. Second, this algorithm is one-sided in the sense that it detects either increases or decreases of the expected value of the signal, depending on the sign of the change in magnitude δ. The simplest solution, already proposed by Page,[42] is to use a two-sided CUSUM algorithm. It is constituted by two one-sided CUSUM algorithms running in parallel and using the same positive change in magnitude δ, one using $+\delta$ to detect an increase, and the other using $-\delta$ to detect a decrease in the expected value of the signal. Third, the algorithm presented in Algorithm 1 stops as soon as an abrupt change is detected. In order to segment multi-level events, it is restarted each time a change is detected, until the end of the event. After these three minor modifications, we obtain a simple and efficient sequential change detection algorithm, which relies on a double-sided CUSUM algorithm and has only two user-defined parameters:

– δ, the positive change in magnitude, corresponding to the magnitude of the most likely encountered changes in the signal which have to be detected quickly,

– h, the positive detection threshold, related to the algorithm sensitivity.

These two parameters directly influence the global performance of the algorithm, which is formally given by its average run length (ARL) function. Page defined this quantity as the expected number of samples before an action is taken when $\mu = \mu_0$:

$$\mathrm{ARL}_0(\delta,h) = E_{\mu 0}[n_\mathrm{d}] \quad (2)$$

In this equation, $E[\ ]$ is the expectation operator and n_d is the detection time of the algorithm. Therefore, this quantity corresponds to the expected number of samples before a false alarm is signaled, and can be viewed as the average time between two false detections, or equivalently as the inverse of a false alarm rate. As mentioned in chapter 2 of the book by Hawkins and Olwell,[46] this ARL function can be used to set the parameters δ and h as follows:

– set δ to the magnitude of the most likely change encountered in the signal,

– choose $\mathrm{ARL}_0(\delta,h)$ as the smallest acceptable number of samples between two false detections,

– determine the value of h required for the two previous chosen values thanks to dedicated codes or tables (see for example chapter 3 of Hawkins and Olwell[46]).

In the context of our application, the following values lead to good segmentation performance:

– change in magnitude: $\delta = 0.2$ nA

– average run length function $\mathrm{ARL}_0 = 500$ samples

By using these two values in the tables given in chapter 3 of this book,[46] we finally obtain a detection threshold $h = 1 \times \frac{\delta}{\sigma}$ with σ the standard deviation of the baseline.

As an example, the proposed segmentation algorithm is applied with the previous settings to a multi-level event, and the corresponding results are shown in Fig. 4c. As can be noticed in this figure, segmentation results obtained with these settings are quite satisfactory. All translocation events presented in this paper are segmented by applying the CUSUM method and above listed settings.

3.4. Event database

Once translocation events have been detected and segmented by the two previous methods, important event information is stored in a dedicated database:

– event nature (impulsive or not),

– event start and end,

– number of levels in the event,

– current values and dwell times of each level.

As shown in the next section, this database can be used to further analyze each event independently, or to classify the different events detected in the current signal regarding their nature, their number of levels, *etc.* All described subroutines are written in MATLAB and are part of the OpenNanopore software package.

4. Results and discussion

Typical fitted results are shown in Fig. 1d. Impulses, one-level events and two-level events are given as examples of events fitted with the CUSUM method. We can clearly see that our program fits all events including impulses. Fig. 5 displays the results of λ DNA translocation through the nanopore and compares data analysis done using two approaches; the point histogram approach is illustrated in grey and the level histogram approach is illustrated in red, green and dark blue. In a point histogram, each point in the histogram is a point in the original signal whereas in a level histogram, each point in the histogram is a level in an event. The level histogram approach is the one developed in this paper, where the event dwell time and the current blockage values are given by the CUSUM algorithm. If there is more than one level within the event then each level has its own dwell time, current blockage and order in which it occurs. In order to identify each sub-population we use a short MATLAB script (also provided in OpenNanopore package) that extracts one, two and multi-level events from the main table. Copies of those events are then reported in separate tables. Using those tables, we can easily work on a sub-population of events. In both level histograms in Fig. 5 it is possible to identify the one-level events in green, the shallow current blockage of the two-level events in blue and the deep current blockage of the two-level events in red. The CUSUM method performs well whether the noise is low (Fig. 5a) or high (Fig. 5b), and it is easy to identify levels from the level histogram; clearly this is not the case for high noise in the point histogram (Fig. 5b).

Another level fitting method has been developed by Storm *et al.*[27] where a point histogram of the recorded events is used to find levels that are more likely to happen (peaks in the histogram). The events are fitted to the closest level found in this histogram. This method is easy to implement but requires some knowledge on the levels prior to fitting, which is not the case for the CUSUM method. In our method the main input parameters are either straightforward or we propose a technique to calculate them easily. Moreover, in the method developed by Storm *et al.*[27] the user is limited to the resolution of the histogram and the levels extracted from the peaks. Levels within three times the standard deviation of each other are not visible in such a point histogram because the populations overlap. Resolving of different populations is hard when two Gaussian distributions have a distance of less than four times the standard deviation between their mean values. For example, in the experiments of Storm *et al.*,[27] the typical separation between peaks is six times the standard deviation.

In order to compare our results to the related study performed by Storm *et al.*,[27] we have calculated the SNR[34] as a measure of how well our method performs. This comparison can easily be made since in both cases the salt concentration was 1 M KCl and the applied voltage was 100 mV. The absolute current blockage due to unfolded DNA translocation is $\Delta I = 0.2$ nA; this is the minimal jump in the mean to detect. The RMS current noise values are given in Section 2.2. Using those values we obtain for the low noise measurement a SNR = 5.42 and for the high noise measurement a SNR = 3.59. If we perform an estimate of the SNR of Storm *et al.*'s[27] measurements, assuming that their distribution is Gaussian, we can evaluate their RMS current noise value to 10 pA which means that they have a SNR = 15 for an absolute current blockage of $\Delta I = 0.15$ nA. With the CUSUM method we can detect and fit events even if the SNR is close to 1 (see ESI† for more details). The lower the SNR, the longer the event has to be in order to be detected.[46]

For impulses (10 samples = 100 μs), the CUSUM algorithm itself either does not detects the impulse or if it detects the impulse, the fit is not accurate. We have created a workaround so that those events are also fitted and listed in the event database and it is part of the OpenNanopore package. Some of those short

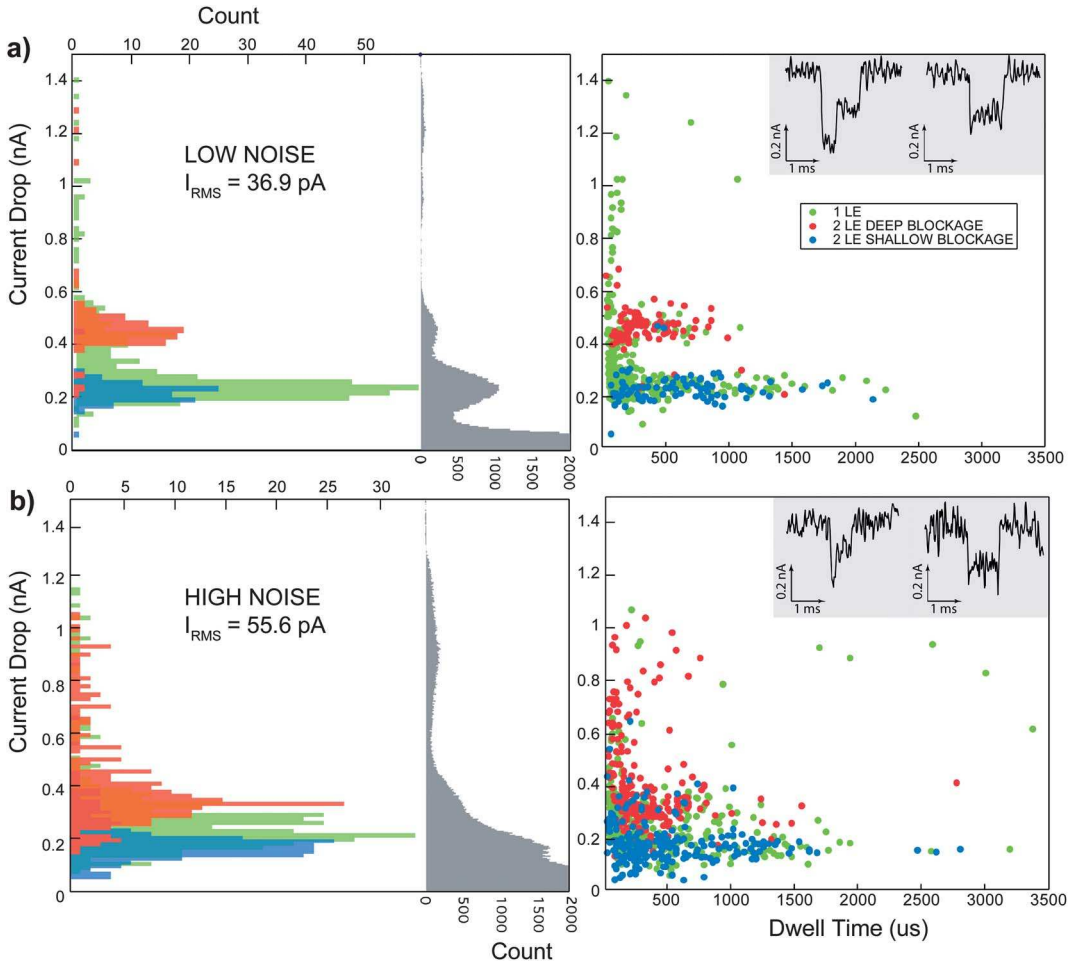

Fig. 5 Scatter plots, point and level histograms of current blockages for (a) low noise (I_{rms} = 36.9 pA at 100 mV at 10 kHz low-pass filter) and (b) high noise measurements (I_{rms} = 55.6 pA at 100 mV at 10 kHz low-pass filter). Scatter plots of one-level events (1LE) in green, deep in red and shallow in blue current blockages of two-level events (2LE). Comparison of point histograms and level histograms with the same color code: point histograms of all events (in grey) on top of level histograms of 1LE, high and low blockages of 2LE.

events are also influenced by the filter; this effect has been thoroughly studied by another group.[47] The OpenNanopore software package runs in 5.2 s per million data points where 3.6 s are due to plotting; more detail on the performance of the software and the minimal impulse length detected by the CUSUM algorithm is given in the ESI.†

The main advantage and novelty of the OpenNanopore package is that it does level detection. Other advantages are that it is very fast, user-friendly, requiring a small number of user-defined parameters, it performs under low SNR, and it is a free and open-source software so it can be upgraded and modified. Its disadvantages are that the input signal must have a stable baseline (there is no pre-processing in OpenNanopore) and it has a limited number of ready-to-use processing functions compared to other software packages. Since our software is based on the widely used MATLAB platform, it is easy with the given output structure (the event database) to implement other post-processing functions.

In conclusion, the CUSUM method was successfully tested on λ DNA translocation experiments in nanopores as well as in nanocapillaries[48] (data not shown). This experimental data was used as a proof of principle experiment to demonstrate the efficiency of this new method even on noisy datasets. The future applications of this method, in combination with experimental adjustments such as translocation in high viscosity solution,[36,49] could be used to detect shorter levels in more complex signals. This new method for detection of levels inside events could also be used to study macromolecules such as protein–DNA complexes.[13] Evolution of the method could be used for smaller molecules and, in the future, for analysis of DNA sequencing experiments done by Derrington *et al.* and Clarke *et al.*[50,51]

All OpenNanopore MATLAB files and a GUI can be downloaded from our laboratory website at lben.epfl.ch.

Acknowledgements

This work was financially supported by the European Research Council (grant no. 259398, PorABEL: Nanopore integrated nanoelectrodes for biomolecular manipulation and sensing). C.R. was financed by a grant from the Swiss SystemsX.ch initiative (IPhD), evaluated by the Swiss National Science Foundation. Nanopore fabrication was carried out at the EPFL Center for Micro/Nanotechnology (CMI) and at the Centre Interdisciplinaire de Microscopic Electronique (CIME). We

would also like to thank S. Quatrefages and E. Garcia-Cordero for working on a first version of the program.

References

1 L. Song, M. Hobaugh, C. Shustak, S. Cheley, H. Bayley and J. Gouaux, *Science*, 1996, **274**, 1859–1866.
2 J. Li, D. Stein, C. McMullan, D. Branton, M. Aziz and J. Golovchenko, *Nature*, 2001, **412**, 166–169.
3 A. Storm, J. Chen, X. Ling, H. Zandbergen and C. Dekker, *Nat. Mater.*, 2003, **2**, 537–540.
4 J. Kasianowicz, E. Brandin, D. Branton and D. Deamer, *Proc. Natl. Acad. Sci. U. S. A.*, 1996, **93**, 13770–13773.
5 J. Li, M. Gershow, D. Stein, E. Brandin and J. Golovchenko, *Nat. Mater.*, 2003, **2**, 611–615.
6 A. Storm, C. Storm, J. Chen, H. Zandbergen, J. Joanny and C. Dekker, *Nano Lett.*, 2005, **5**, 1193–1197.
7 A. Meller, L. Nivon, E. Brandin, J. Golovchenko and D. Branton, *Proc. Natl. Acad. Sci. U. S. A.*, 2000, **97**, 1079–1084.
8 M. Wanunu, T. Dadosh, V. Ray, J. Jin, L. Mcreynolds and M. Drndić, *Nat. Nanotechnol.*, 2010, 1–8.
9 R. I. Stefureac, D. Trivedi, A. Marziali and J. S. Lee, *J. Phys.: Condens. Matter*, 2010, **22**, 454133.
10 D. S. Talaga and J. Li, *J. Am. Chem. Soc.*, 2009, **131**, 9287–9297.
11 M. Wanunu, J. Sutin and A. Meller, *Nano Lett.*, 2009, **9**, 3498–3502.
12 R. M. M. Smeets, S. W. Kowalczyk, A. R. Hall, N. H. Dekker and C. Dekker, *Nano Lett.*, 2009, **9**, 3089–3095.
13 C. Raillon, P. Cousin, F. Traversi, E. Garcia-Cordero, N. Hernandez and A. Radenovic, *Nano Lett.*, 2012, **12**, 1157–1164.
14 M. Wanunu, S. Bhattacharya, Y. Xie, Y. Tor, A. Aksimentiev and M. Drndic, *ACS Nano*, 2011, **5**, 9345–9353.
15 S. M. Iqbal, *Nanopores*, Springer Verlag, 2011.
16 M. Wanunu, J. Sutin, B. McNally, A. Chow and A. Meller, *Biophys. J.*, 2008, **95**, 4716–4725.
17 M. van den Hout, V. Krudde, X. J. A. Janssen and N. H. Dekker, *Biophys. J.*, 2010, **99**, 3840–3848.
18 B. Lu, F. Albertorio, D. P. Hoogerheide and J. A. Golovchenko, *Biophys. J.*, 2011, **101**, 70–79.
19 G. Skinner, M. van den Hout, O. Broekmans, C. Dekker and N. Dekker, *Nano Lett.*, 2009, **9**, 2953–2960.
20 R. F. Purnell, K. K. Mehta and J. J. Schmidt, *Nano Lett.*, 2008, **8**, 3029–3034.
21 D. Stoddart, A. J. Heron, E. Mikhailova, G. Maglia and H. Bayley, *Proc. Natl. Acad. Sci. U. S. A.*, 2009, **106**, 7702–7707.
22 F. Qin, A. Auerbach and F. Sachs, *Biophys. J.*, 2000, **79**, 1915–1927.
23 F. Qin and L. Li, *Biophys. J.*, 2004, **87**, 1657–1671.
24 A. Churbanov and S. Winters-Hilt, *BMC Bioinf.*, 2008, **9**, S13.
25 S. Winters-Hilt, W. Vercoutere, V. S. Deguzman, D. Deamer, M. Akeson and D. Haussler, *Biophys. J.*, 2008, **84**, 967–976.
26 B. Konnanath, P. Sattigeri, T. Mathew, A. Spanias, S. Prasad, M. Goryll, T. Thornton, and P. Knee, *19th International Conference on Artificial Neural Networks (ICANN 2009)*, 2009, 5769, pp. 265–274.
27 A. Storm, J. Chen, H. Zandbergen and C. Dekker, *Phys. Rev. E: Stat., Nonlinear, Soft Matter Phys.*, 2005, **71**, 051903.
28 T. Osaki, J. P. Barbot, R. Kawano, H. Sasaki, O. Français, B. L. Pioufle and S. Takeuchi, *Procedia Eng.*, 2010, **5**, 796–799.
29 Z. Siwy and A. Fulinski, *Phys. Rev. Lett.*, 2002, **89**, 158101.
30 P. Chen, T. Mitsui, D. Farmer, J. Golovchenko, R. Gordon and D. Branton, *Nano Lett.*, 2004, **4**, 1333–1337.
31 D. P. Hoogerheide, S. Garaj and J. A. Golovchenko, *Phys. Rev. Lett.*, 2009, **102**, 256804.
32 S. W. Kowalczyk, A. Y. Grosberg, Y. Rabin and C. Dekker, *Nanotechnology*, 2011, **22**, 315101.
33 V. Tabard-Cossa, D. Trivedi, M. Wiggin, N. Jetha and A. Marziali, *Nanotechnology*, 2007, **18**, 305505.
34 R. M. M. Smeets, U. F. Keyser, N. H. Dekker and C. Dekker, *Proc. Natl. Acad. Sci. U. S. A.*, 2008, **105**, 417–421.
35 R. M. M. Smeets, U. F. Keyser, M. Y. Wu, N. H. Dekker and C. Dekker, *Phys. Rev. Lett.*, 2006, **97**, 088101.
36 D. Fologea, J. Uplinger, B. Thomas, D. McNabb and J. Li, *Nano Lett.*, 2005, **5**, 1734–1737.
37 U. Mirsaidov, J. Comer, V. Dimitrov, A. Aksimentiev and G. Timp, *Nanotechnology*, 2010, **21**, 395501.
38 D. C. Montgomery, *Statistical Quality Control: a Modern Introduction*, Wiley, 6th edn, 2010.
39 H. M. Wadsworth, K. S. Stephens, and A. B. Godfrey, *Modern Methods for Quality Control and Improvement*, John Wiley & Sons, 2nd edn, 2008.
40 M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*, Prentice-Hall Inc., 1993.
41 T. L. Lai, *J. Roy. Statist. Soc. Ser. B*, 1995, **57**, 613–658.
42 E. S. Page, *Biometrika*, 1954, **41**, 100–114.
43 G. Lorden, *Ann. Math. Stat.*, 1971, **42**, 1897–1908.
44 G. V. Moustakides, *Ann. Stat.*, 1986, **14**, 1379–1387.
45 Y. Ritov, *Ann. Stat.*, 1990, **18**, 1464–1469.
46 D. M. Hawkins and D. H. Olwell, *Cumulative Sum Charts and Charting for Quality Improvement*, Springer Verlag, 1998.
47 D. Pedone, M. Firnkes and U. Rant, *Anal. Chem.*, 2009, **81**, 9689–9694.
48 L. J. Steinbock, O. Otto, D. R. Skarstam, S. Jahn, C. Chimerel, J. L. Gornall and U. F. Keyser, *J. Phys.: Condens. Matter*, 2010, **22**, 454113.
49 J. Li and D. S. Talaga, *J. Phys.: Condens. Matter*, 2010, **22**, 454129.
50 I. M. Derrington, T. Z. Butler, M. D. Collins, E. Manrao, M. Pavlenok, M. Niederweis and J. H. Gundlach, *Proc. Natl. Acad. Sci. U. S. A.*, 2010, **107**, 16060–16065.
51 J. Clarke, H.-C. Wu, L. Jayasinghe, A. Patel, S. Reid and H. Bayley, *Nat. Nanotechnol.*, 2009, **4**, 265–270.