Supporting Information

1) Experimental Conditions and Nanopore Fabrication

To fabricate nanopores we use the well established e-beam drilling technique[1]. The nanopore used in the two sets of experiments was fabricated using a Philips/FEI CM300 TEM operated at 200 kV, further detail on nanopore fabrication can be found elsewhere[2]. After fabrication, the pore was stored in a degassed and filtered 1:1 ddH_2O:EtOH solution until use[3]. The nanopore chip was then mounted in a PMMA microfluidic cell and the two reservoirs on each side of the pore were filled with 0.22 um filtered and degassed buffer containing 1 M KCl, 1 mM Tris/HCl pH 7.5, and 0.1 mM EDTA, we used silicon o-rings to create a good seal between the two chambers. The Ag/AgCl electrodes connected to the pre-amplifier of the Axopatch were immersed in the two reservoirs and a bias voltage of 100 mV was applied. This part of the setup was mounted on a damping table (Thorlabs, NJ) and enclosed in a Faraday cage. Signals were filtered using a 4-pole lowpass Bessel filter at a cut-off frequency of 10 kHz. Signals were sampled at 100 kHz using a National Instrument PXI-4461 DAQ card.

2) Power Spectral Densities

The current standard deviation is evaluated in the frequency domain by taking the square root of the Power Spectral Density (PSD) summed over the full bandwidth up to 50 kHz. The values found in the frequency domain are in good agreement with the values found in the time domain, thus demonstrating a correct calibration of the PSD. Both PSD have been computed using Welch's averaged modified periodogram method applied to data without events. The total data length used for PSD estimation was 217965 samples while the data segments were 16384 samples long with 75% overlap, allowing a good convergence of

spectrum estimates. The FFT size was 4 * 16384 padded with zero, giving a frequency resolution Δf of about 1.5 Hz up to 50 kHz.

In Fig. 2 we see very clearly the slope of the low-pass Bessel filter effect, which is at -60 dB/decade. In fact the Bessel filter is a 4-pole order filter, which is then -80 dB/decade slope. This demonstrates that the high frequency current noise spectrum increasing 20 dB/decade is still present up to the maximum frequency analysis of 50 kHz. The measured slope being then -80 + 20 = -60 dB/decade. The PSD of our low noise and high noise measurements are comparable to the ones shown in *Tabard-Cossa et al.*[4] without the PDMS layer whether cleaning with piranha has been made or not.

3) Event detection using adaptive thresholds

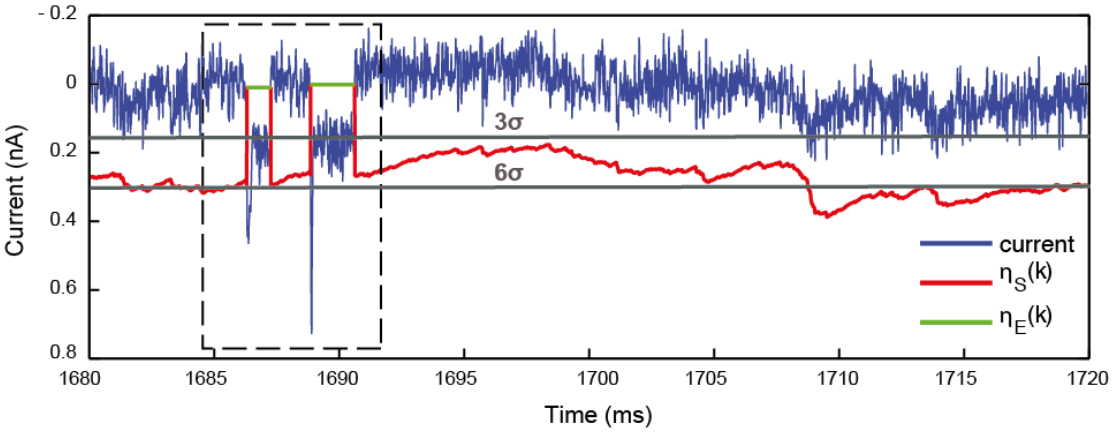

Figure SI-1. Event detection done with adaptive thresholds and comparison with classical thresholds (3 to 6 σ).

The goal of the event detection step is to detect and roughly localize eventual translocation events in the measured current. The approach usually applied to detect a translocation event is to use a threshold. This is implemented by comparing the current sample value $i(k)$ to a threshold $\eta_S = \mu - S\sigma$, where μ and σ are the mean value and standard deviation of the current, and S is a positive parameter set by the user. A translocation event is then simply detected at this sample if $i(k)$ is lower than η_S. As shown

in the previous section the presence of a $1/f$ noise in the measured current signal can vary significantly. This type of noise contains strong low frequency components, which can drastically increase the false detection rate of the threshold method.

One way to overcome this problem is to calculate an adaptive threshold, which automatically adapts to the local quantity of very low frequency components. In the event detection method proposed here, this is realized by defining a local threshold $\eta_S(k)$ through local estimates of the mean $\mu(k)$ and standard deviation $\sigma(k)$ of the current signal as follows:

$$\eta_S(k) = \mu(k) - S\sigma(k) \qquad (2)$$

The local mean value $\mu(k)$ is estimated by applying a first-order low-pass recursive filter to the current $i(k)$ as defined in Eq.(3).

$$\mu(k) = a\mu(k-1) + (1-a)i(k) \qquad (3)$$

In this equation, a is the only parameter of this filter and should be smaller and closer to 1 in order to obtain a stable low-pass filter. The local standard deviation $\sigma(k)$ is estimated by applying the same filter to $\left(i(k) - \mu(k)\right)^2$ and by taking the square root of the filter's output as shown in Eq. (4).

$$\sigma^2(k) = a\sigma^2(k-1) + (1-a)\left(i(k) - \mu(k)\right)^2$$
$$\sigma(k) = \sqrt{\sigma^2(k)} \qquad (4)$$

The next sample of the current $i(k+1)$ is then compared to the adaptive threshold $\eta_S(k)$ defined in Eq. (2) in order to decide if a translocation event starts at this sample. Once the start of an event has been detected, its end is localized at the first next sample whose value returns above a second threshold $\eta_E(k)$. This threshold is defined as $\eta_E(k) = \mu(k) - E\sigma(k)$, but with a positive parameter $0 \le E < S$ such that $\eta_S(k) < \eta_E(k) \le \mu(k)$.

4) Minimal Impulse Length detected by the CUSUM algorithm

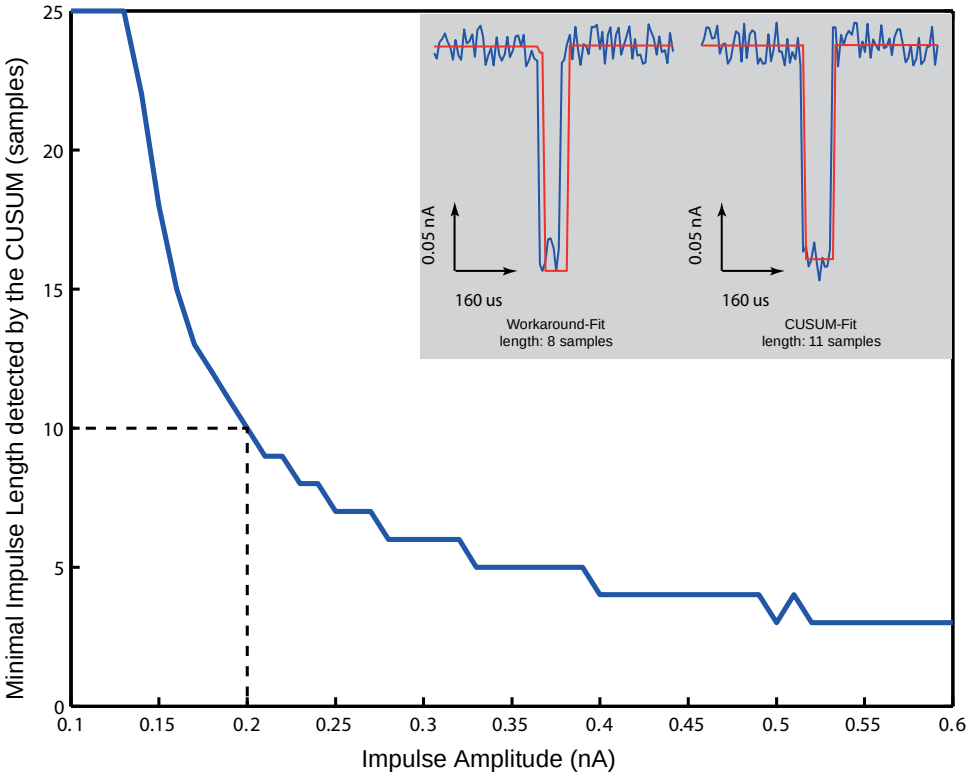

Figure SI-2. Graph of minimal impulse length detected by the CUSUM algorithm versus the impulse amplitude at fixed CUSUM settings.

Fig. SI-2 shows the minimum number of samples of an impulse detected by the CUSUM as a function of its current blockage (artificial signals). We can see that the minimal length an impulse must have in order to be detected by the CUSUM depends highly on the amplitude of its current blockage. In the scope of this article, the CUSUM was fine-tuned so that the current blockage that is optimally detected is the one of a single folded λ DNA strand, i.e. setting $\delta = 0.2$ nA. The graph above supports the theoretical conclusions about CUSUM performance drawn in the article. Current blockages smaller than δ must be quite long to be detected by the CUSUM. This curve can be used to find the threshold where we can begin to use the CUSUM algorithm to detect short impulsions. At an impulse amplitude of 0.2 nA, the minimum impulse length that can be detected by the CUSUM algorithm is 10 samples.

The inlets of Fig. SI-2 show a fit done by the short-impulse workaround (left current trace) and a fit of an impulse detected by the CUSUM (right current trace).

5) Performance of the software package

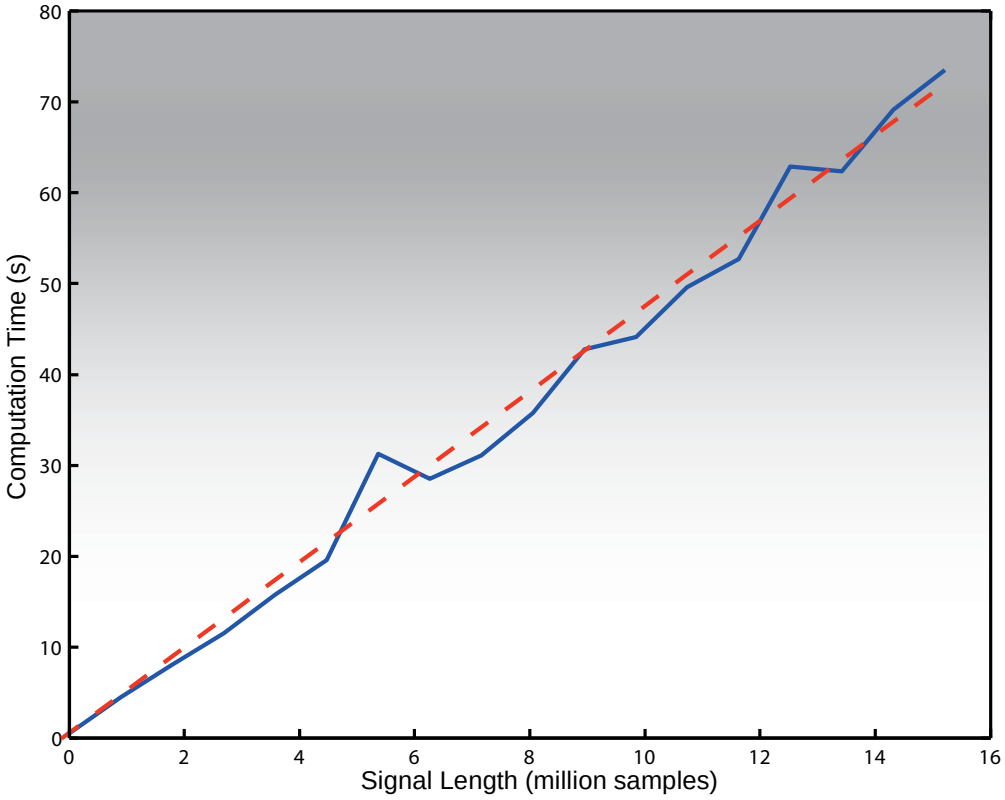

Figure SI-3. Performance of the software package. This graph shows the linear behavior of the OpenNanopore software package up to 16 million samples. The total speed was 210 kS/s (kilo samples per second) where 13% is taken by the filter, 18% by the event detection and 69% by the plotting. The MATLAB version used for this work was R2011b, the processor was an Intel Core i7 with 4 cores, the memory was 4 GB and the operating system was Mac OS X version 10.7.3.

6) Noise influence on CUSUM algorithm performance

Goal:

The goal is to study the influence of noise on the performance of the fitting step done through the CUSUM algorithm.

Methodology:

The general methodology is to measure the performance of the fitting method by analyzing the results obtained on a synthetic signal with different noise quantities. This synthetic signal has the following model:

$$i[n] = i_{event}[n] + i_n[n], \quad \textbf{(5)}$$

where $i_{event}[n]$ is a piecewise constant signal modeling a typical event, and $i_n[n]$ is an additive centered white Gaussian noise with standard deviation σ.

In our application, a typical event is multi-level, with a current drop between two consecutive levels of at least δ = 0.2 nA, and a dwell time of about 50 ms per level. The event signal $i_{event}[n]$ is generated so as to verify these assumptions:

- two-level event,
- first level: dwell time of 50 ms, value of -0.2 nA,
- second level: dwell time of 50 ms, value of -0.4 nA.

Moreover, 50 ms of baseline (0 nA) is added on each side of this chosen event to finally form $i_{event}[n]$.

The signal to noise ratio SNR compares the quantity of additive noise $i_n[n]$ with the importance of the event signal $i_{event}[n]$, and is defined as $SNR = \frac{\delta}{\sigma}$. Fig. SI-4 a) and b) shows two noisy synthetic signals $i[n]$, one with a "high" $SNR = 4$ (on the left) and one with a

"low" $SNR = 1$ (on the right). The event signal $i_{event}[n]$ is represented in dashed line on the same figure.

Fitting results:

Once applied on the noisy signal $i[n]$, the output of the CUSUM algorithm is a fitted signal $i_{fit}[n]$, which consists of a piecewise constant signal trying to estimate the event signal $i_{event}[n]$. Fig. 2 shows the results obtained by the CUSUM algorithm on noisy signals with the same SNR as in Fig. SI-4 a) and b), with the fitted signal $i_{fit}[n]$ represented in red.

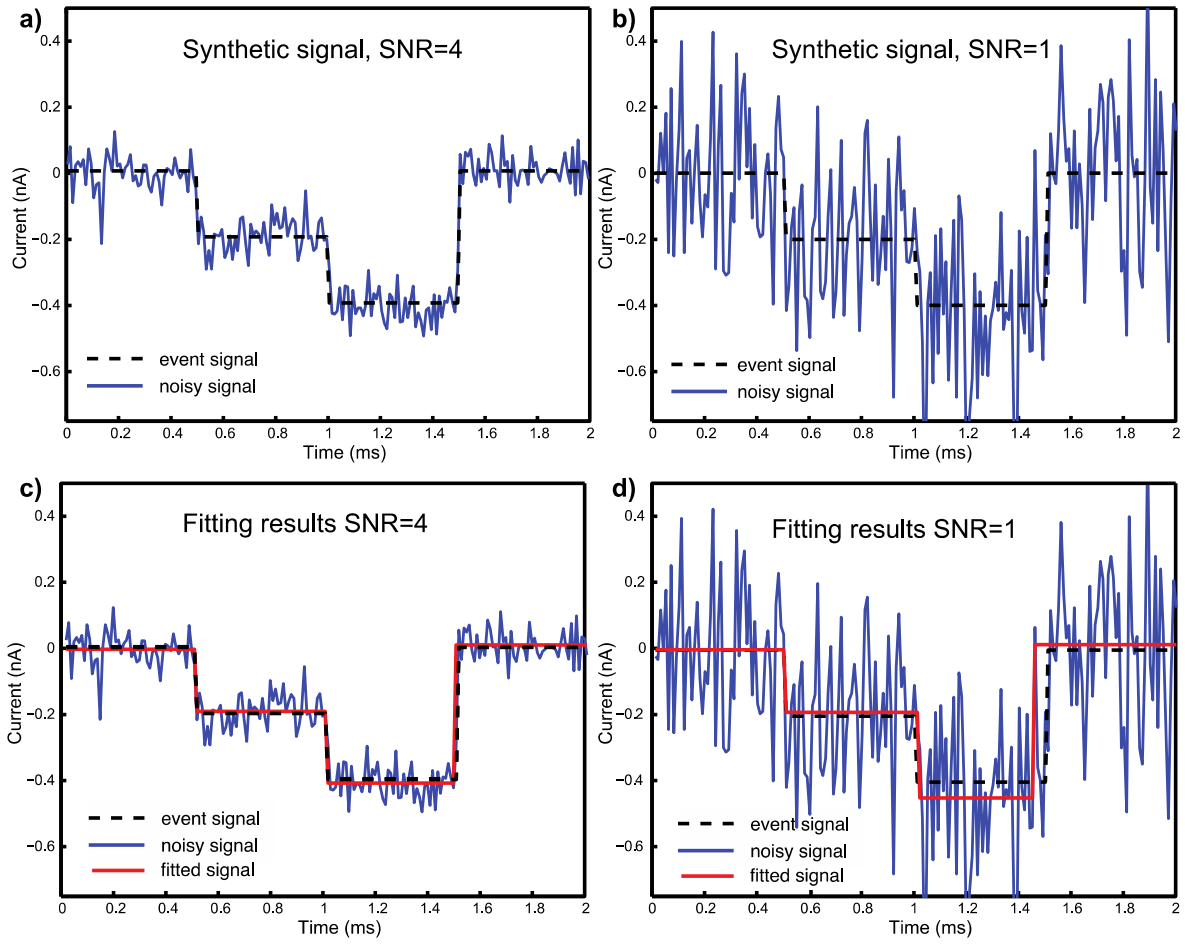

Figure SI-4. Influence of noise on the performance of the fitting step done with the CUSUM algorithm. a) and b) Noisy synthetic signals with high and low SNR respectively. c) and d) Results of fit with high and low SNR respectively.

Performance criterion:

The measure of the performance is done thanks to the error of fit E_{fit}. This quantity is the square root of the normalized mean squared error between the desired event signal $i_{event}[n]$ and the fitted signal $i_{fit}[n]$:

$$E_{fit} = \sqrt{\frac{\sum_{n}\left(i_{event}[n] - i_{fit}[n]\right)^2}{\sum_{n}\left(i_{event}[n]\right)^2}} \qquad (6)$$

Clearly, this quantity can be interpreted as the normalized distance between the two previous signals $i_{event}[n]$ and $i_{fit}[n]$. As an example, the value obtained for E_{fit} in the high SNR case of Fig. SI-4 c) is 4%, and increases to 33% in the low SNR case .of Fig. SI-4 d).

In order to obtain significant statistical results, 1000 loops are realized for each SNR, and the average value of these 1000 errors is calculated to finally obtain the average error of fit $\overline{E}_{fit}$. This quantity is the global criterion of fit used in the following, and is obviously close to 0% in the case of a good fit, and close to 100% in the worst case.

Results:

Fig. SI-5 shows in logarithmic and linear scale the behavior of the global criterion of fit $\overline{E}_{fit}$ in % with respect to SNR. The SNR ranges from 0.1 (lot of noise) to 100 (little noise).

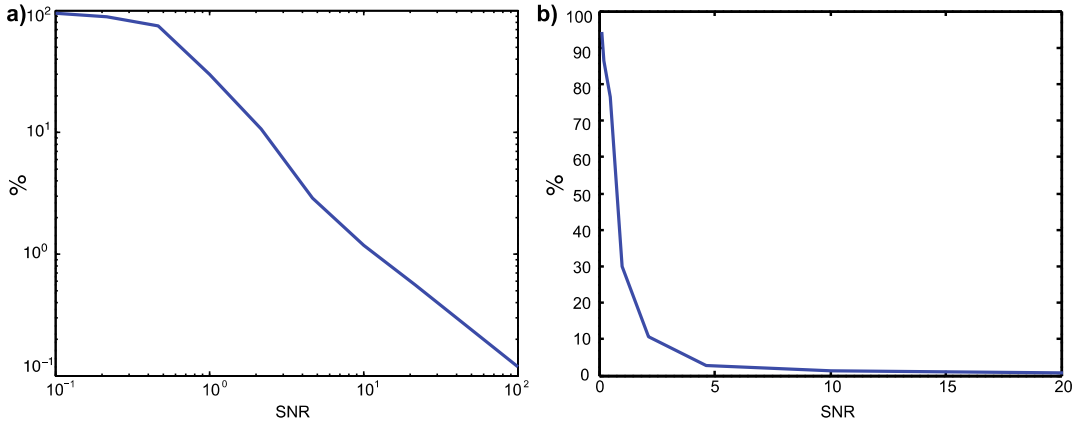

Figure SI-5. a) and b) Average error of fit $\overline{E}_{fit}$ as a function of SNR in logarithmic and linear scale respectively.

The logarithmic scale curve clearly shows that if the noise quantity is very important (SNR lower than 0.5), the fitting performance is bad and the average error of fit stays close to its maximum value of 100%. For lower noise quantities and higher SNR, the error progressively decreases as the SNR increases above 0.5, to finally reach the acceptable value of 10% error around $SNR = 2$ and very small errors for higher SNR. The linear scale curve highlights the rapid performance improvement as the SNR linearly increases.

Conclusion:

As a conclusion, this statistical study shows that the proposed fitting method applied to a typical event reaches acceptable performance when the SNR reaches 1, and that this performance becomes excellent for SNR higher than 2.

REFERENCES:

1. A. Storm, J. Chen, X. Ling, H. Zandbergen, and C. Dekker, *Nature Materials*, 2003, **2**, 537–540.
2. C. Raillon, P. Cousin, F. Traversi, E. Garcia-Cordero, N. Hernandez, and A. Radenovic, *Nano Lett.*, 2012, **12**, 1157–1164.
3. M. van den Hout, V. Krudde, X. J. A. Janssen, and N. H. Dekker, *Biophysical Journal*, 2010, **99**, 3840–3848.
4. V. Tabard-Cossa, D. Trivedi, M. Wiggin, N. Jetha, and A. Marziali, *Nanotechnology*, 2007, **18**.