# Automatic Analysis of Large-scale Nanopore Data Using Hidden Markov Models

**Jianhua Zhang*, Xiuling Liu****

**Department of Computer Science, Oslo Metropolitan University, 0166 Oslo, Norway (e-mail: jianhuaz@oslomet.no)*
*** School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237 P.R. China (e-mail:1437242295@qq.com)*

**Abstract:** In this paper we developed a modified Hidden Markov Model (HMM) to analyze the raw nanopore experimental data. Traditionally, prior to further analysis the measured nanopore data must be pre-filtered, but the filtering usually distorts the waveform of the blockage current, especially for rapid translocations and bumping blockages. The HMM is known to be robust with respect to strong noise and thus suitable for processing the raw nanopore data, but its performance is susceptible to the setting of initial parameters. To overcome this problem, we use the Fuzzy c-Means (FCM) algorithm to initialize the HMM parameters in this work. Then we use the Viterbi training algorithm to optimize the HMM. Finally, both the simulated and experimental data analysis results are presented to show the effectiveness of the proposed method for detection of the nanopore current blockage events in analytical chemistry.

***Keywords:*** Nanopore; Time series analysis; Hidden Markov model; Viterbi algorithm; Fuzzy c-means clustering algorithm.

## 1. INTRODUCTION

The basic principle of nanopore analysis technique is that a molecule passes the nanopore resulting in a temporary reduction in the ionic current and through analyzing the current amplitude and time duration of the blocked current signal, we can identify the biochemical information of analyte. It has been widely used for single-molecule detection of ion, DNA, RNA, protein and peptide (Braha et al., 2000; Kasianowicz et al., 1996; Ying et al., 2011; Movileanu et al., 2005). Recent work (for example Ashton et al., 2015; Loose et al., 2016; and Quick et al., 2016) revealed that the nanopore analysis is able to accurately sequence virus and bacterial pathogens.

The ionic current signal measured from the nanopore experiment is inevitably corrupted by noises. Therefore, the short blockage events with low current amplitude are easily buried in noise and hard to detect. To facilitate signal detection and processing, it is necessary to remove the higher-frequency noise using a low-pass filter. However, the low-pass filter usually distorts the signals especially for the short events with duration shorter than $2*T_r$ , where $T_r = 0.3321/f_c$ is the rising time of the filter with $f_c$ its cut-off frequency (Gu et al., 2015). It increases the duration and reduces the current amplitude, which makes it difficult to determine the molecules' biochemical information accurately.

Two strategies have been commonly used to mitigate the above issues induced by pre-filtering: 1) retrieve the distorted events based on the method of equivalent electric circuit of nanopore (Balijepalli et al. (2014)), Full-width-half-maximum (FWHM) (Arjmandi et al. (2012); Plesa *et al.* (2015)), the slope of event (Pedone et al. (2009)), or the area of event (Gu et al., 2015) to improve the performance of pre-filtered data analysis; 2) Improve the experimental equipment performance (Garalde et al. 2013; O'Donnell et al. 2012), such as increase the bandwidth of experimental equipment to reduce the degree of signal distortion, but this would introduce strong noise and make traditional methods no longer applicable. In addition, due to the high degree of system integration, parts of the experimental device become easily consumable, which highly increases the detection cost.

In order to overcome the disadvantages of the above two strategies, we directly process the highly noisy raw (unfiltered and almost undistorted) nanopore experimental data based on the Hidden Markov Model (HMM) to detect the current blockage events' biochemical information. The HMM (Rabiner et al. (1989); Dugad et al. (1996)) is tolerant of the strong noise and thus has been successfully used to detect events from the noisy ionic current signal measured by patch-clamp (Chung et al. (1990); Chung et al. (1991); Qin et al. (2004)). Unfortunately, the HMM is sensitive to its initial parameters usually pre-set manually in previous work, which is not suited to implementation of automatic data processing. In this work, we utilize the Fuzzy c-means (FCM) clustering algorithm to initialize the parameters of HMM for practical applications.

## 2. EXPERIMENTS

### *2.1 Materials*

α-Hemolysin (α-HL) wildtype-D8H6 was produced by expression in BL21 (DE3) pLysS Escherichia coli cells and self-assembled into heptamers, and decane were purchased from Sigma-Aldrich (⩾99%, St. Louis, MO, USA). 1, 2-Diphytanoyl-sn-glycero-3-phosphocholine (chloroform, ⩾

99%) was purchased from Avanti Polar Lipids (Alabaster, AL). All oligonucleotides used in our experiments were synthesized by Invitrogen Life Technologies (Shanghai, China). Ultrapure water (resistivity of 18.2 MΩ•cm at 25 °C) was obtained from a Milli-Q system (EMD Millipore, Billerica, MA). The pH 8.0 buffer solution used was composed of 1 M KCl.

### *2.2 Experimental Procedure*

As described in (Ying et al. (2013), Ying et al. (2011) and Liu Y et al. (2013), the lipid bilayers were created by applying 1,2-diphytanoyl-sn-glycero-3-phosphocholine (30 mg/mL) in decane (⩾99%, Sigma-Aldrich, St. Louis, MO, USA) to a 150 µm orifice in a 1 mL bilayer chamber (Warner Instruments, Hamden, CT, USA) filled with KCl (1.0 M) and Tris-HCl (10 mM, pH = 8.0). The stability of the bilayer was evaluated by monitoring its resistance and capacitance. The solution of α-hemolysin was injected into the *cis* chamber proximal to the bilayer. Then seven monomers of α-hemolysin assembled to form a hydrophilic channel in the bilayer. The two compartments of the bilayer cell are termed *cis* and *trans*. A pair of Ag/AgCl electrodes was inserted into the two compartments. After a single nanopore was formed on the bilayer, the analyte was injected into the *cis* chamber. The voltage was set to +100 mV during the experiments. A ChemClamp instrument (Dagan Co., Minneapolis, MN) in the voltage clamp mode was used to amplify and measure the ionic current flowing through the nanopore. The filtered and unfiltered data were measured simultaneously at a sampling rate of 100 kHz by using a DigiData 1440A A/D converter (Axon Instruments, Forest City, CA, USA) and the filter cut-off frequency is 3 kHz. Data was recorded by the PClamp software (Axon Instruments).

## 3. METHODS

### *3.1 Data Analysis Procedure*

The data processing procedure developed is shown in Fig. 1.

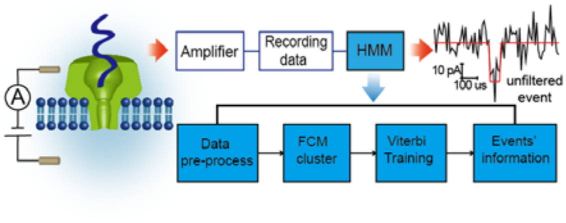

Fig. 1. The data analysis procedure.

The HMM has been successfully used in the single-channel current signal recorded by Patch-clamp, which is similar to the nanopore data (Chung et al. (1990); Qin et al. (2004)). It can be assumed that nanopore current signal is generated by a 1st-order discrete-time finite-state Markovian process with Gaussian white noise, but the state of the process (submerged in noise) is not directly observable (or measurable). Therefore, the nanopore data can be modelled by the HMM with the observable current data ($O_1$, $O_2$,⋯,$O_T$) and the hidden (unobservable) state sequence ($q_1$, $q_2$,⋯, $q_T$).

The HMM, applied to nanopore data analysis, consists of the following components/parameters (Rabiner et al. (1989)):

1. The observation (observed sample) sequence $O_1, O_2, \cdots, O_T$ with the length of $T$. For the nanopore data analysis problem, the observations correspond to the measured current data.

2. The set of hidden states $S = \{S_1, S_2, \cdots, S_N\}$, where $N$ is the cardinality of the set, say the number of hidden states. The observed sample $O_t$, $t$=1, 2,⋯,$T$ can be generated by several hidden state $q_t \in S$ with certain probability. The observation sequence $O_1$, $O_2$,⋯,$O_T$ usually corresponds to multiple hidden state sequences $q_1$, $q_2$,⋯, $q_T$, and we call the most likely hidden state sequence optimal in the sense of maximum likelihood. In the nanopore problem, the hidden states correspond to the $N$ current levels in the current signal.

3. The $N \times N$ state transition probability matrix $A = \{a_{ij}\}$, where $a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i)$, $1 \le i, j \le N$ denotes the state of the HMM at time $t$. In the nanopore problem, the transition probability denotes the probability of a transition from current level $S_i$ to $S_j$.

4. The initial state distribution $\boldsymbol{\pi} = \{\pi_i\}$, where $\pi_i = P(q_1 = S_i), 1 \le i \le N$ is $N$-dimensional column vector. In the nanopore problem, it denotes the probability that the first observed sample $O_1$ results from each current level.

5. The observation probability distribution matrix in the state $S_j : B = \{b_j(O_t)\}$, where $O_t$ is the observation at time $t$, and $b_j(O_t) = P(O_t \mid q_t = S_j), 1 \le j \le N$. In the nanopore problem, the probability distribution of the observed data due to the state $S_i$ is assumed to be Gaussian.

The HMM is usually used to solve the following three typical problems:

**Problem 1**. Given the model $\lambda = (\boldsymbol{\pi}, A, B)$, determine the occurrence probability $P(O \mid \lambda)$ of observation sequence $O_1, O_2, \cdots, O_T$. The typical method for this problem is Forward and Backward algorithm (Devijver et al. (1985); Rabiner et al. (1990)).

**Problem 2**. Given the model $\lambda = (\boldsymbol{\pi}, A, B)$ and observation sequence $O_1, O_2, \cdots, O_T$, find the optimal state sequence $q_1, q_2, \cdots, q_T$ to maximize the probability $P(O, S \mid \lambda)$. The typical method for this problem is Viterbi algorithm, which will be briefly introduced later on (Forney et al. (1973); Rabiner et al. (1990)).

**Problem 3**. Adjust the parameters in the model $\lambda = (\boldsymbol{\pi}, A, B)$ such that the probability $P(O \mid \lambda)$ is maximized. There are two typical methods to optimize the HMM parameters: the Viterbi training algorithm (aka. segmental $k$-means in some literature) (Bhowmik et al. (2011); Juang et al. (1990); Rabiner et al. (1990)) and Baum-Welch algorithm.

As mentioned above, in the nanopore data analysis problem under study, the observations correspond to the current sample data; the hidden states correspond to the current blockage events (stairs), $N$ denotes the number of current blockage events in the current signal, the transition probability is the probability of state transition from current level $S_i$ to $S_j$ , and the probability distribution of the observation belonging to state $S_i$ is assumed to be Gaussian distribution $N(\mu_i, \sigma_i^2)$ , where $\mu_i$ is the mean of sample belonging to $S_i$ and $\sigma_i^2$ is the variance (Chung et al. (1990); Qin et al. (2004)). The probability of observation $O_t$ generated by $S_i$ can be calculated by:

$$b_i(O_t) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{(O_t - \mu_i)^2}{2\hat{\sigma}_i^2}\right] . \qquad (1)$$

The task is to assign each sample data $O_t$ to the corresponding current level $S_i$ by using the HMM to remove the noise and estimate the current events. In other words, we want to accurately estimate the information of current blockage events, such as the amplitude and duration of each current level by means of the Viterbi algorithm.

### *3.2 Data Preprocessing*

In a long raw experimental data, we are interested in the detection of those current blockage events only. Therefore, in order to improve the computational efficiency of our algorithm, we set a small threshold and find the data points in the current time-series signal whose current amplitudes are smaller than the threshold. Then we only need to process these data points using our algorithm.

### *3.3 Initialization of HMM Parameters*

Before using the Viterbi training algorithm to optimize the HMM parameters, a set of initial parameters must be set, including the initial state distribution probability vector **π**, state transition probability matrix $A$, and probability matrix of observation $B$ consisting by the means and variances of each state. In most cases, the initial parameters **π** and $A$ have little influence on the results, hence these two parameters can be set randomly or fixed. However, the initial value of $B$ usually has significant effect on the result. It was found by Qin et al. (2004) and Hu et al. (2011) that the Viterbi training algorithm is more sensitive to the means $\mu$ and variances $\sigma^2$ than to state transition probabilities. Thus it is important to use an appropriate method to estimate the initial mean $\mu$ and variance $\sigma^2$ of each state.

Here we use clustering algorithm to initialize the HMM parameters. The most commonly-used clustering algorithm is *k*-means (Likas et al. (2003)), but it is very sensitive to the initial cluster centroid and its performance could be affected if the data of different classes have obvious overlapping. Fuzzy *c*-means algorithm (Bezdek et al. (1984); Pal et al. (1995)) is an improved version of the *k*-means algorithm. In the *k*-means algorithm, each sample belongs to each cluster with a probability of either 0 or 1, while in fuzzy clustering, each data point belongs to each cluster with a membership degree between 0 and 1. When the data of different classes overlap severely, the performance of FCM is more stable than *k*-means and traditional hierarchical clustering algorithms (Mingoti et al. (2006)).

In our problem the samples of neighbouring current levels often overlap, so we choose the FCM algorithm to initialize the HMM parameters. More specifically, we firstly cluster the observations to obtain the class label for each data point using the FCM algorithm, then based on the clustering results we can determine the initial value of **π**, $A$, and $B$ according to (11)-(15).

The FCM algorithm partitions a set of observation $O_1, O_2, \cdots, O_T$ into several clusters by minimizing the objective function:

$$J_m = \sum_{j=1}^{T} \sum_{i=1}^{N} w_{ij}^m \left\| O_j - \mu_i \right\|^2 \quad ,1 \le m < \infty \ . \qquad (2)$$

Where $T$ is the number of samples, $N$ the number of clusters (i.e., the number of current levels in our problem), $m$ the fuzziness parameter, $\mu_i$ the center of the $i$ -th cluster, $w_{ij} \in (0,1)$ the degree of membership of data $O_j$ in the $i$ -th cluster, and $\|*\|$ denotes the Euclidean distance.

The previous work showed that the weighting exponent $m$ greatly influences the FCM performance. For instance, Bezdek *et al.* (1984) stated that the value of $m$ controls the degree of samples shared by different clusters. Pal and others (1995) examined effect of the parameter $m$ on cluster validity and found that the optimum range of $m$ is [1.5, 2.5]. Therefore, we set the value of $m$ as 2.

Fuzzy partitioning is carried out through an iterative optimization of the objective function defined in (2). The membership degrees and cluster centres $\mu_i$ are updated by

$$\begin{cases} w_{ij} = \dfrac{1}{\sum_{k=1}^{N} \left( \dfrac{\left\| O_j - \mu_i \right\|}{\left\| O_j - \mu_k \right\|} \right)^{\frac{2}{m-1}}} \\ \mu_i = \dfrac{\sum_{j=1}^{T} w_{ij}^m \cdot O_j}{\sum_{j=1}^{T} w_{ij}^m} \end{cases} \qquad (3)$$

Bezdek *et al.* (1984) showed that the numerical convergence of FCM algorithm can usually be achieved in 10-25 iterations. So in our problem, the algorithm iteration is terminated if the variation of membership degree matrix is less than $\varepsilon = 0.0001$ or the maximum number of iterations $k = 100$ is reached. The iterative procedure of the FCM algorithm consists of the following computational steps:

**Step 1**: Initialize membership degree matrix $W = \left[ w_{ij} \right]$, where $w_{ij} \in (0,1)$.

**Step 2**: At the *k*-th iteration, calculate the class centre $\mu^k = [\mu_i]$ and update $W^k$ by using (3), where $W^k$ denotes the membership degree matrix in the *k*-th iteration.

**Step 3**: If $J_m^k - J_m^{k-1} < \varepsilon$ or $k > 100$, terminate the algorithm; Otherwise loop back to **Step 2**.

### *3.4 Optimization of HMM Parameters*

Given the initial model parameters and a set of observed data, the Viterbi or Baum-Welch algorithm can be used to optimize the HMM parameters. However, the two algorithms are quite different. In the Baum-Welch algorithm, the model parameters $\lambda = (\boldsymbol{\pi}, A, B)$ are tuned until $P(O \mid \lambda)$ (the probability of the observation sequence $O$ generated by model $\lambda$) is maximized. In the Viterbi algorithm, the model parameters $\lambda = (\boldsymbol{\pi}, A, B)$ are tuned until the probability $P(O, S \mid \lambda)$ (the probability of the observation sequence $O$ generated by model $\lambda$ and the optimal state sequence $S$) is maximized. The Viterbi training algorithm only considers the best possible state sequence when tuning the model parameters in the iterative process, while the Baum-Welch algorithm is a full-likelihood approach by summing up the probabilities of all possible state sequences and thereby produces better estimates of model parameters. However, the Viterbi algorithm is usually preferred because we are mostly interested in the occurrence of the observation sequence from the best possible state sequence. Moreover, the Viterbi algorithm requires much less computation than the Baum-Welch algorithm and has confirmed nice performance in practical applications (Rodríguez et al. (2003); Allahverdyan et al. (2011)). In the sequel, we will make a detailed comparison between the two methods on the simulated data.

### *3.5 Viterbi Training Algorithm*

The Viterbi algorithm is briefly introduced here. For a more complete description of the algorithm, the interested readers are referred to Dugad *et al.* (1996).

To estimate the optimal state sequence $q_1, q_2, \cdots, q_t$ from observation sequence $O_1, O_2, \cdots, O_t$, we define the maximum probability along a single path at time $t$ which accounts for the first $t$ observations by the hidden state $S_i$ as:

$$\delta_t(i) = \max_{q_1, q_2, \cdots, q_{t-1}} P(q_1 q_2 \cdots q_t = S_i, O_1 O_2 \cdots O_t \mid \lambda) \tag{4}$$

then we have

$$\delta_{t+1}(j) = \max_{1 \le i \le N} [\delta_t(i) a_{ij}] b_j(O_{t+1}) \tag{5}$$

Moreover, we use $\psi_t(j)$ to indicate the state that maximized $\delta_t(j)$. The procedure of finding the best state sequence can be summarized as follows.

**Step 1 -** Initialization:

$$\begin{aligned} &\delta_1(i) = \pi_i b_i(O_1), 1 \le i \le N \\ &\psi_1(i) = 0 \end{aligned} \tag{6}$$

**Step 2 -** Recursion:

$$\begin{aligned} \delta_t(j) &= \max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), 2 \le t \le T, 1 \le j \le N \\ \psi_t(j) &= \arg\max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}], \quad 2 \le t \le T, 1 \le j \le N \end{aligned} \tag{7}$$

**Step 3 -** Termination:

$$\begin{aligned} P^* &= \max_{1 \le i \le N} [\delta_T(i)] \\ q_T^* &= \arg\max_{1 \le i \le N} [\delta_T(i)] \end{aligned} \tag{8}$$

**Step 4 -** Path (state sequence) backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \cdots 1. \tag{9}$$

The input arguments of the Viterbi algorithm are $\lambda = (\boldsymbol{\pi}, A, B)$ and the observation sequence and the output argument is the estimated class label of each observed data. Given a nanopore blocking current time-series signal $O_1, O_2, \cdots, O_t$, we use the Viterbi algorithm to classify each data point into several classes (i.e., hidden states in HMM) and to detect the blocking current events. More specifically, given an observation sequence and the initial HMM model λ, we classify each data point through the Viterbi algorithm. Based on the classification results obtained, we re-calculate the initial probabilities **π**, transition probabilities *A* and the probability distribution matrix of observation *B*.

If the first data point's class label is $i$, we can determine

$$\boldsymbol{\pi}(i) = 1, 1 \le i \le N \tag{10}$$

According to Qin *et al.* (2004), the transition probabilities *A* can be determined by

$$a_{ij} = \frac{n(i,j)}{n(i)}, \ 1 \le i \le N, 1 \le j \le N \tag{11}$$

where $n(i,j)$ is the number of occurrences of $\left\{ O_t \in S_i \ and \ O_{t+1} \in S_j \right\}$ for all $t$ and $n(i)$ the number of occurrences $\left\{ O_t \in S_i \right\}$ for all $t$.

We recalculate mean and variance of each current level by:

$$\begin{cases} \mu_i = \dfrac{\sum\limits_{O_t \in S_i} O_t}{\sum\limits_{O_t \in S_i} 1} \\ \sigma_i^2 = \dfrac{\sum\limits_{O_t \in S_i} (O_t - \mu_i)^2}{\sum\limits_{O_t \in S_i} 1} \end{cases} \tag{12}$$

Then we determine $B$ using $\mu_i$, $\sigma_i$ and the Gaussian p.d.f.:

$$b_i(O_t) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[ -\frac{(O_t - \mu_i)^2}{2\sigma_i^2} \right] \tag{13}$$

The training procedure continues iteratively until the variation in the probability $P(O,S\,|\,\lambda)$ falls within a pre-set threshold (set as 0.0001 in our data analysis). The flowchart of the Viterbi training algorithm is shown in Fig. 2.

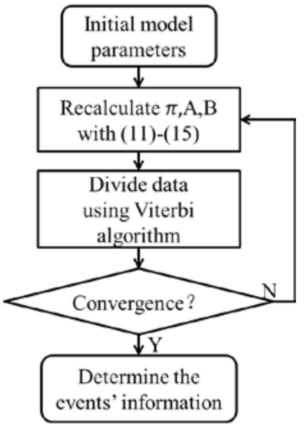
Initial model parameters
Recalculate π,A,B with (11)-(15)
Divide data using Viterbi algorithm
Convergence?
N
Y
Determine the events' information

Fig. 2. Flowchart of the Viterbi training algorithm.

## 4. RESULTS AND DISCUSSION

### *4.1 Synthetic Data*

In order to validate the performance of the proposed method on the raw (unfiltered) signal, we firstly apply it on the simulated blockage current data. The frequency of the simulated data is 100 kHz, a total of 500 ms data were generated involving 800 short blockage current events with the duration of 70-130 μs. The baseline current is 3 pA and the amplitude of blocking current follows a Gaussian distribution with the mean and variance of 2 pA and 0.1 pA, respectively. Then zero-mean Gaussian white noise with s.d. ranging from 0.1 pA to 0.5 pA (with an interval of 0.1 pA) was added to generate five trials of simulated data, whose SNR *i*/σ=10, 5, 3.3, 2.5, and 2 respectively.

Firstly FCM algorithm, with two clusters (i.e., current levels), is used to process the simulated data. For example, when σ=0.3 pA (this noise level is very close to the experimental data), a sample (column 1-3, …, 31186, 31187 …, 49998-50000) of the membership degree matrix (2×50000), obtained by FCM algorithm, is:

$$\begin{pmatrix} 0.01 & 0.05 & 0.02 & \cdots & 0.93 & 0.53 & \cdots & 0.03 & 0.01 & 0.06 \\ 0.99 & 0.95 & 0.98 & \cdots & 0.07 & 0.47 & \cdots & 0.97 & 0.99 & 0.94 \end{pmatrix}$$

Based on this matrix, the cluster label of each data point can be determined by using the maximum membership degree approach.

Then the initial parameter of HMM is determined by using (10)-(13) and the samples' class labels. The initial parameters determined by FCM are:

$$\boldsymbol{\pi}=\begin{pmatrix} 1 \\ 0 \end{pmatrix},\ \mathrm{A}=\begin{pmatrix} 0.88 & 0.12 \\ 0.37 & 0.63 \end{pmatrix}$$
$$\boldsymbol{\mu}=\begin{pmatrix} 3.05 \\ 2{,}14 \end{pmatrix},\ \boldsymbol{\sigma}=\begin{pmatrix} 0.26 \\ 0.34 \end{pmatrix}$$

Then we use the Viterbi training algorithm to optimize these parameters, obtaining the optimized parameters:

$$\boldsymbol{\pi}=\begin{pmatrix} 1 \\ 0 \end{pmatrix},\ \mathrm{A}=\begin{pmatrix} 0.98 & 0.02 \\ 0.10 & 0.90 \end{pmatrix}$$
$$\boldsymbol{\mu}=\begin{pmatrix} 3.00 \\ 2{,}00 \end{pmatrix},\ \boldsymbol{\sigma}=\begin{pmatrix} 0.30 \\ 0.31 \end{pmatrix}$$

By comparison we can find that the values of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ before and after optimization are very close. Then we use the class labels obtained to determine the time duration of each event. Fig. 3(A) shows the simulated data under five different levels of noise and the restored signal acquired by our method (in red line). Then the simulated signals were filtered by using a 5 kHz low-pass Bessel filter (in this case, the event with time duration shorter than 130 μs will be severely distorted). The filtered signals were processed by the FWHM method (Arjmandi et al., 2012) and DBC (2nd-order Differential-Based Calibration) method (Gu et al., 2015).

To compare the three methods, we make a statistical analysis of the time duration obtained by each of them. The too short event with time duration shorter than 40 μs (i.e., comprising 4 or less data points) is excluded from the statistical analysis. Fig. 3(B) shows the statistical distribution of the time duration data acquired by the three methods under different levels of noise. The histogram of the (known) true duration of simulated data is shown in Fig. 4(A). Compared with the results on the filtered data obtained by the FWHM and DBC method, the results on the unfiltered obtained by HMM are closer to the true value (70-130 μs).

To evaluate the accuracy of the three methods, we define the mean relative error (MRE) as:

$$MRE = \frac{1}{n}\sum_{i=1}^{n} (\hat{t}_i - t_{\mathrm{norm}}) / t_{\mathrm{nom}} \tag{14}$$

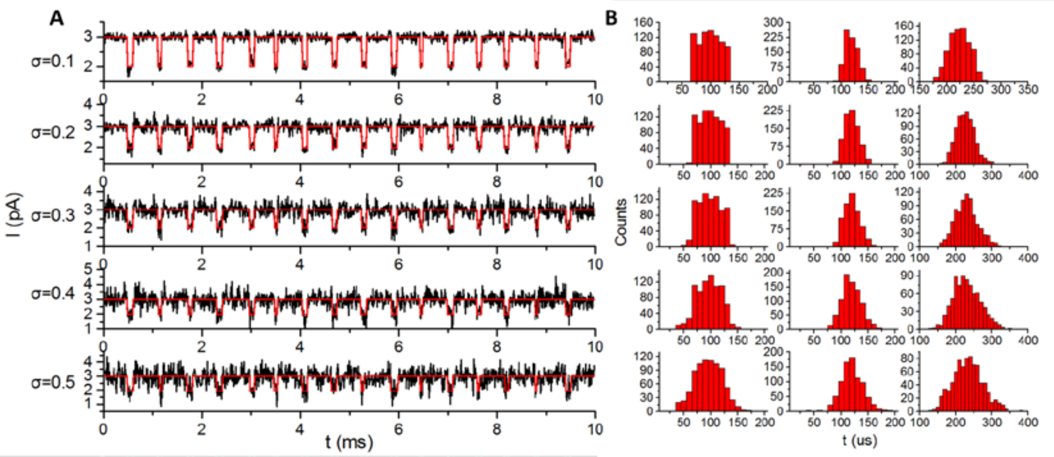

Fig. 3. (A) Simulated data (unfiltered) with five levels of noise (~1000 samples shown) and the results of our method (red line); (B) The distribution of event's duration (~800 events): (left) HMM; (middle) FWHM applied on the filtered data (5 kHz low-pass Bessel filter); (right) DBC applied on the filtered data.

where $n$ is the number of the events with the same duration in the simulated data, $\hat{t}_i$ the estimated duration of the $i$-th event, and $t_{nom}$ the true value (70-130 µs).

Fig. 4(B-F) compares the MRE of duration acquired by the three methods, from which we can find that under all five different levels of noise the proposed method resulted in the least MRE. For the strongest noise with i/σ = 2 (σ = 0.5 pA), the MRE of the proposed method is about 20%, which is much smaller than that of DBC (~55%) and FWHM method (~45%). Therefore, more accurate information can be extracted from the unfiltered data by the modified HMM method (MHMM).

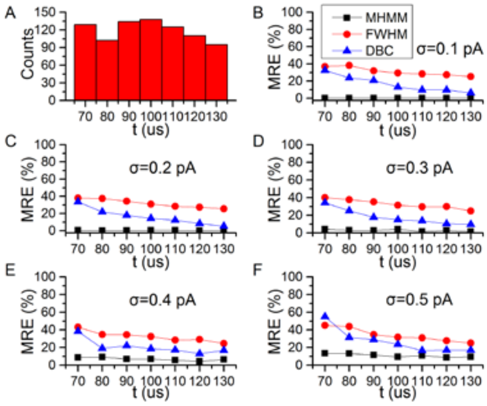

Fig. 4. Comparison of mean relative error of event's duration of the three methods under five different levels of noise.

### *4.2 Comparison of Viterbi and Baum-Welch Algorithms*

To further substantiate the superiority of the Viterbi training algorithm for our problem, we compare the performance of the Viterbi and Baum-Welch algorithms in this section.

We evaluated the two methods in terms of the error rate of the number of the detected events (Qin et al. (2004)), defined by $|n - n_{nom}| / n_{nom}$ with $n_{nom}$ (~800) being the true number of events set in the simulations and $n$ the estimated number events detected. Fig. 5(A) shows the error rate of the two algorithms under 5 different levels of noise, from which we can find that both algorithms attained a low error rate and that the error rate of both algorithms increases for the noisier data. This is because with the increase of the noise level, it becomes more difficult to separate the interested current blocking events from the noise and when the noise is very strong, some noise may be mixed up with the short events. In general, the performance of the Baum-Welch and Viterbi algorithm is comparable.

Furthermore, we also make an analysis of the time duration time of each detected event. The duration of the simulated events is in the range of 70-130 µs. The MRE of events' duration is calculated by using (14).

Fig. 5(B) shows the MRE results on the simulated data with event duration of 70 µs, 90 µs, 110 µs and 130 µs under five levels of additive noise. We can find that the error of average duration of the detected events increases with higher level of noise. With the same level of noise, the shorter events have a higher error rate because they are submerged in noise.

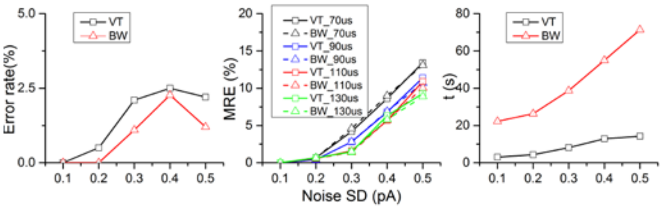

Fig. 5. (left) The error rate of the detection of the number of events; (middle) MRE of event's duration. Four types of events with the duration of 70 µs, 90 µs, 110 µs, and 130 µs are shown in black, blue, red and green, respectively. The solid and dotted line denotes the results of Viterbi and Baum-Welch algorithm, respectively; (right) Computational time.

In addition, we compare the computational efficiency (or time consumption) of the two algorithms coded and run on the computing environment of Intel(R) Core(TM) i5-2450 CPU @2.5GHz, 4G RAM, 64 bit Win7 Prof. OS and Matlab

R2013a. The time consumption of the two algorithms under five levels of noise is shown in Fig. 5(C). We can find the computational time required by both algorithms increases with higher level of noise. This is mainly because for noisier data, they need more iteration to converge. Furthermore, we found that with the same level of noise, the Baum-Welch algorithm is much slower than the Viterbi algorithm (it takes nearly 5 times of computational time required by the latter).

Moreover, the Viterbi algorithm can achieve the mean amplitude of current blocking events. For example, when $\sigma = 0.3$ pA (SNR $\approx 3.3$), The two levels' amplitude derived by the Viterbi training algorithms are 1.99 pA and 3 pA, which is very close to the true values.

From the performance of the Viterbi training algorithm on the simulated data, we may conclude that it is very tolerant of the noise and thus suitable for processing the raw (unfiltered) nanopore experimental data with lower signal-to-noise ratio. By comparing the Viterbi and Baum-Welch algorithms, we found that both algorithms can detect the current blocking events accurately but the former is more efficient computationally especially for the challenging problem of large-scale data analysis in real time.

### *4.3 Experimental Data*

In this section, we applied our method to the unfiltered experimental data of poly(dA)$_{30}$ of about 100 s. A sample of the results (~ 70 ms) are shown in Fig. 6(A). Fig. 6(B) shows five sample events detected by our method. We also use the existing FWHM and DBC method to the corresponding filtered data by using a 3 kHz low-pass Bessel filter. Table 1 presents the five events' time duration time estimated by the three methods.

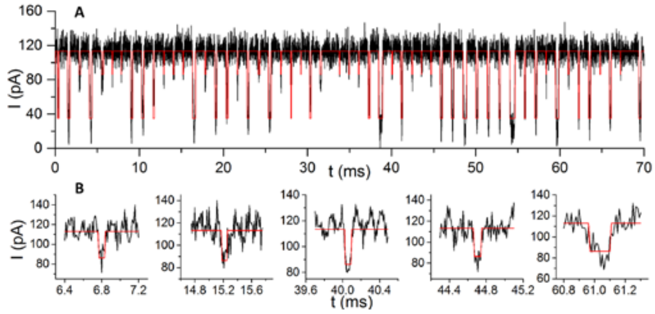

Fig. 6. (A) The prefiltered experimental data of poly(dA)$_{30}$ (~70 ms data) and results of our method (red line); (B) a sample of unfiltered signal and the events detected by our method.

Table 1. Comparison of the event duration estimated by the three methods.

| Method | Event duration estimated (μs) | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| HMM | 70 | 80 | 70 | 80 | 140 |
| FWHM | 170 | 140 | 110 | 150 | 160 |
| DBC | 250 | 250 | 200 | 260 | 290 |

The histograms of the events' time duration estimated by the three methods are compared in Fig. 7. All the histograms are fitted by a Gaussian function. The time duration obtained by our method on the unfiltered data is $0.13 \pm 0.008$ ms, while the results of FWHM and DBC methods are $0.17 \pm 0.003$ ms and $0.33 \pm 0.003$ ms, respectively.

The significant difference in time duration between the proposed method and the two existing methods is mainly due to the unwanted filtering effect on many very short events: the increase of the time duration in general. The comparative results demonstrated the capacity of the proposed method for accurate and efficient elicitation of events from the raw experimental data.

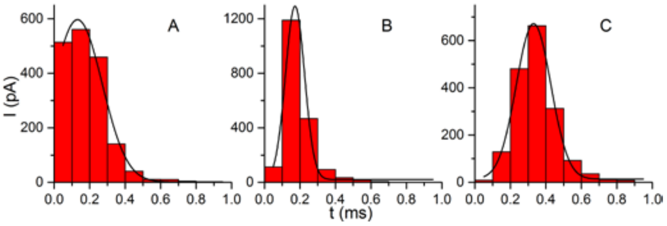

Fig. 7. The duration histogram of poly(dA)$_{30}$: (A) HMM applied on raw data; (B) FWHM applied on the filtered data; (C) DBC. The statistical results were fitted by the Gaussian function (black line).

## 5. CONCLUSION

In this paper, in order to alleviate the sensitivity of HMM to its initial parameters setting, we utilized the FCM clustering technique to initialize the HMM parameters. Then we used the modified HMM to process nanopore experimental data. The analysis results of both the simulated and experimental raw nanopore data showed that the proposed method is more accurate than traditional methods for detection of the current blocking events. The Viterbi training algorithm is shown to be faster than the Baum-Welch algorithm by a factor of about 5 and thus more suitable for online data analysis. Furthermore, the proposed method is shown to be especially suited for short current blockage events, which are hard to accurately detect by traditional methods due to the signal distortion by pre-filtering.

## REFERENCES

Allahverdyan, A. and Galstyan, A. (2011). Comparative analysis of Viterbi training and maximum likelihood estimation for hmms. In *Advances in Neural Information Processing Systems*, 1674-1682.

Arjmandi, N., Van Roy, W., Lagae, L., and Borghs, G. (2012). Improved algorithms for nanopore signal processing. *arXiv preprint arXiv*:1207.2319.

Ashton, P. M., Nair, S., Dallman, T., Rubino, S., Rabsch, W., Mwaigwisya, S. and O'Grady, J. (2015). MinION nanopore sequencing identifies the position and structure of a bacterial antibiotic resistance island. *Nature Biotechnology*, 33(3), 296-300.

Balijepalli, A., Ettedgui, J., Cornio, A. T., Robertson, J. W., Cheung, K. P., Kasianowicz, J. J., and Vaz, C. (2014).

Quantifying short-lived events in multistate ionic current measurements. *ACS nano*, 8(2), 1547-1553.

Bezdek, J. C., Ehrlich, R., and Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers and Geosciences*, 10(2-3), 191-203.

Bhowmik, T. K., van Oosten, J. P., and Schomaker, L. (2011). Segmental K-means learning with mixture distribution for HMM based handwriting recognition. In *Proc. of Int. Conf. on Pattern Recognition and Machine Intelligence* (pp. 432-439). Berlin; Heidelberg: Springer, June 2011.

Braha, O., Gu, L. Q., Zhou, L., Lu, X., Cheley, S., and Bayley, H. (2000). Simultaneous stochastic sensing of divalent metal ions. *Nature Biotechnology*, 18(9), 1005-1007.

Chung, S. H., Krishnamurthy, V., and Moore, J. B. (1991). Adaptive processing techniques based on hidden Markov models for characterizing very small channel currents buried in noise and deterministic interferences. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 334(1271), 357-384.

Chung, S. H., Moore, J. B., Xia, L., Premkumar, L. S., and Gage, P. W. (1990). Characterization of single channel currents using digital signal processing techniques based on hidden Markov models. *Philosophical Transactions of the Royal Society of London B: Biological Sciences,* 329(1254), 265-285.

Devijver, P. A. (1985). Baum's forward-backward algorithm revisited. *Pattern Recognition Letters*, 3(6), 369-373.

Dugad, R., and Desai, U. B. (1996). *A Tutorial on Hidden Markov Models*. Report No.: SPANN-96.1, Signal Processing and Artificial Neural Networks Lab, Dept. of Electrical Engineering, Indian Institute of Technology, Bombay.

Forney, G. D. (1973). The Viterbi algorithm. *Proc. of IEEE*, 61(3), 268-278.

Garalde, D. R., O'Donnell, C. R., Maitra, R. D., Wiberg, D. M., Wang, G., and Dunbar, W. B. (2013). Modeling the biological nanopore instrument for biomolecular state estimation. *IEEE Trans. on Control Systems Technology*, 21(6), 2038-2051.

Gu, Z., Ying, Y. L., Cao, C., He, P., and Long, Y. T. (2015). Accurate data process for nanopore analysis. *Analytical Chemistry*, 87(2), 907-913.

Hu, L., and Zanibbi, R. (2011). HMM-based recognition of online handwritten mathematical symbols using segmental k-means initialization and a modified pen-up/down feature. In *Proc. of IEEE Int. Conf. on Document Analysis and Recognition*, pp. 457-462, Sep. 2011.

Juang, B. H., and Rabiner, L. R. (1990). The segmental K-means algorithm for estimating parameters of hidden Markov models. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 38(9), 1639-1641.

Kasianowicz, J. J., Brandin, E., Branton, D., and Deamer, D. W. (1996). Characterization of individual polynucleotide molecules using a membrane channel. *Proc. of the National Academy of Sciences*, 93(24), 13770-13773.

Likas, A., Vlassis, N., and Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern Recognition*, 36(2), 451-461.

Liu, Y., Ying, Y. L., Wang, H. Y., Cao, C., Li, D. W., Zhang, W. Q., and Long, Y. T. (2013). Real-time monitoring of the oxidative response of a membrane–channel biomimetic system to free radicals. *Chemical Communications*, 49(59), 6584-6586.

Loose, M., Malla, S., and Stout, M. (2016). Real time selective sequencing using nanopore technology. *BioRxiv*, 038760.

Mingoti, S. A. and Lima, J. O. (2006). Comparing SOM neural network with Fuzzy c-means, K-means and traditional hierarchical clustering algorithms. *European J. of Operational Research*, 174(3), 1742-1759.

Movileanu, L., Schmittschmitt, J. P., Scholtz, J. M., and Bayley, H. (2005). Interactions of peptides with a protein pore. *Biophysical J.*, 89(2), 1030-1045.

O'Donnell, C. R., Wiberg, D. M., and Dunbar, W. B. (2012). A Kalman filter for estimating nanopore channel conductance in voltage-varying experiments. In *Proc of 51st IEEE Annual Conference on Decision and Control (CDC)*, pp. 2304-2309.

Pal, N. R., and Bezdek, J. C. (1995). On cluster validity for the fuzzy c-means model. *IEEE Trans. on Fuzzy Systems*, 3(3), 370-379.

Pedone, D., Firnkes, M., and Rant, U. (2009). Data analysis of translocation events in nanopore experiments. *Analytical Chemistry*, 81(23), 9689-9694.

Plesa, C., and Dekker, C. (2015). Data analysis methods for solid-state nanopores. *Nanotechnology*, 26(8), 084003.

Qin, F. (2004). Restoration of single-channel currents using the segmental k-means method based on hidden Markov modeling. *Biophysical J.*, 86(3), 1488-1501.

Quick, J., Loman, N. J., Duraffour, S., Simpson, J. T., Severi, E., Cowley, L. and Ouédraogo, N. (2016). Real-time, portable genome sequencing for Ebola surveillance. *Nature*, 530(7589), 228-232.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2), 257-286.

Rabiner, L. R., Wilpon, J. G., & Juang, B. H. (1986). A segmental k-means training procedure for connected word recognition. *AT&T Technical J.*, 65(3), 21-31.

Rodríguez, L. J., and Torres, I. (2003). Comparative study of the Baum-Welch and Viterbi training algorithms applied to read and spontaneous speech recognition. In *Proc. of Iberian Conf. on Pattern Recognition and Image Analysis*, pp. 847-857, June 2003.

Ying, Y. L., Li, D. W., Li, Y., Lee, J. S., and Long, Y. T. (2011). Enhanced translocation of poly(dt)$_{45}$ through an α-hemolysin nanopore by binding with antibody. *Chemical Communications*, 47(20), 5690-5692.

Ying, Y. L., Wang, H. Y., Sutherland, T. C., and Long, Y. T. (2011). Monitoring of an ATP‐binding aptamer and its conformational changes using an α-hemolysin nanopore. *Small*, 7(1), 87-94.

Ying, Y. L., Zhang, J., Meng, F. N. *et al.* (2013). A stimuli-responsive nanopore based on a photoresponsive host-guest system. *Scientific Reports*, 3.