Intelligent Decoding of Multi-Level Nanopore Data for Accurate Detection of Cancer Biomarkers

Jian-Hua Zhang,*[a] Xiu-Ling Liu,[a] Zheng-Li Hu,[b] Yi-Lun Ying,*[b] and Yi-Tao Long [b]

[a] School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, P.R. China.

[b] Key Laboratory for Advanced Materials & School of Chemistry and Molecular Engineering, East China University of Science and Technology, Shanghai 200237, P. R. China.

*E-mails: zhangjh@ecust.edu.cn; yilunying@ecust.edu.cn

Electronic Supplementary Information (ESI)

Experimental Section

α-hemolysin (α-HL) was purchased from Sigma-Aldrich (St. Louis, MO, USA) and used without further purification. EDTA and decane (anhydrous, ≥ 99%) were purchased from Sigma-Aldrich (St. Louis, MO, USA). 1,2-Diphytanoyl-sn-glycero-3-phosphocholine (chloroform, ≥ 99%) was purchased from Avanti Polar Lipids Inc. (Alabaster, AL, USA). The microRNA21 with sequence of 5'-UAGCUUAUCAGACUGAUGUUGA-3' was synthesized and HPLC-purified by TaKaRa Biotechnology Co. Ltd. (Dalian, China). The DNA probe 21 with sequence of 5'-TTTTTTTTTTTTTTTTTTTTTCAACATCAGTCTGATAAGCTATTTTTTTTTTTTTTTTTTT-3' was synthesized and HPLC-purified by Sangon Biotech Co., Ltd (Shanghai, China). All reagents and materials were of analytical grade. All solutions were prepared with ultrapure water (18.2 MΩ cm at 25°C) using a Milli-Q System (EMD Millipore, Billerica, MA, USA). The ultrapure water used in the preparation of microRNA21 was treated with DEPC.

All experiments were carried out at 24±2°C. The formation of lipid bilayer and α-HL pore was described in our previous work. [1,2] Both compartments of bilayer apparatus were contained 1 mL of 1 M KCl, 10 mM Tris, 1.0 mM EDTA, pH 8.0. The compartments are termed *cis* and *trans*. The *cis* compartment was connected to the virtual ground. The potential was applied at 140 mV from the *trans* side by an Ag/AgCl electrode. Once a stable single pore was inserted into the bilayer, the analyte was added to the *cis* solution, proximal to the aperture.

In the real sample detection, the detected microRNA was from the original serum. We dissolved probe 21 with DEPC water to 100 μM, premixed the original serum with 10 μL*100 μM probe 21, heated the mixture at 95 ℃ for 3 minutes, and then cooled at 55 ℃ for 3 minutes (this annealing process was performed for 2~3 times). Finally, the annealed mixture of original serum and probe 21 was added to the *cis* chamber and detected.

Currents were amplified under voltage-clamp condition using a ChemClamp instrument (Dagan Co., Minneapolis, MN, USA), filtered by a low-pass Bessel filter with cut-off frequency set at 3 kHz, and digitized at a sampling rate of 100 kHz through a DigiData 1440A converter (Axon Instruments, Forest City, CA, USA) with a PC running PClamp 10.2 (Axon Instruments, Forest City, CA, USA). Each level in the experimental data was manually identified by Clampfit software (Axon Instruments, Forest City, CA, USA).

HMM and nanopore data analysis problem

The HMM has been successfully applied to the single-channel current recording[3,4]. It is assumed that the nanopore current data is generated by a 1st-order, finite-state, discrete-time, Markovian process with unobservable state (submerged in noise) and additive Gaussian white noise. Then it can be modelled by the HMM with the observed current data ($O_1,O_2,\cdots,O_T$) and hidden (unobservable) state sequence ($q_1,q_2,\cdots, q_T$).

The correspondence between the nanopore data analysis problem and HMM [5] parameters is as follows:

1. The observation sequence $O= O_1,O_2,\cdots,O_T$ denotes a sequence of data with the length of T. In the nanopore problem, the observations correspond to the current data.

2. The hidden states $S=\{S_1,S_2,\cdots,S_N\}$, where N is the number of finite hidden states. The observed data O_{t}, $t=1,2,\cdots,T$ can be generated by several hidden state $q_t\in S$ with certain probability. We call the most likely (with maximum likelihood) hidden state sequence optimal. In the nanopore problem, the hidden states correspond to the current levels and N denotes the number of current levels in the nanopore event.

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

4. The initial state distribution $\boldsymbol{\pi}=\{\pi_i\}$, where $\pi_i=P(q_t=S_i)$, $1\leq i\leq N$ is a N-dimensional column vector. In the nanopore problem, it denotes the probability that the first observed data O_1 belongs to each of the current levels.

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 $b_j(O_t)= P(O_t \mid q_t=S_i)$, $1\leq j\leq N$. In the nanopore problem, the distribution of the observation in state S_i is assumed to be a Gaussian distribution $N(\mu_i, \sigma_i^2)$, where μ_i is the mean of the data belonging to hidden state S_i and σ_i^2 its variance.[1,2] The probability of observation O_t generated by S_i can be calculated by:

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

The following three typical problems are related to a HMM:

Problem 1. Given the HMM $\lambda=(\boldsymbol{\pi}, A, B)$, how to determine the occurrence probability $P(O|\lambda)$ of observation sequence $O_1,O_2,\cdots,O_T$,? This problem can be solved by using the Forward and Backward algorithm. [6,7]

Problem 2. Given the HMM $\lambda=(\boldsymbol{\pi}, A, B)$ and observation sequence $O_1,O_2,\cdots,O_T$, how to find the optimal state sequence such that the probability $P(O,S|\lambda)$ is maximized? This problem is typically solved by using the Viterbi algorithm.[8]

Problem 3. How to adjust the parameters in the HMM $\lambda=(\boldsymbol{\pi}, A, B)$ such that the probability $P(O|\lambda)$ is maximized? There are two typical methods to optimize the HMM parameters: One is the Viterbi training algorithm (aka. segmental k-means) [9,10] and another is the Baum-Welch algorithm.[5]

The nanopore data analysis problem is to recover the blockages by assigning each data point to the most probable current level (hidden state), i.e., to estimate the optimal state sequence of the observation sequence. A typical solution to this problem is the Viterbi algorithm. Another problem is related to the optimization of HMM parameters, which can be solved by the Viterbi algorithm as well. The parameter optimization procedure is described in detail as follows.

Viterbi algorithm and Viterbi training procedure

Two algorithms, viz., Baum-Welch and Viterbi training algorithm, have been widely used for the HMM parameter optimization. Many previous studies showed that the Baum-Welch algorithm usually produces a better performance than the Viterbi training algorithm, but its computation complexity is much higher than the latter. In our previous work,[11] we made a comprehensive performance comparison between the Viterbi and Baum-Welch algorithms using the simulated nanopore data. The results showed that the two algorithms achieved comparable accuracy, but the Viterbi training is faster than the Baum-Welch algorithm by a factor of about 5 and thus more suitable for online nanopore data analysis. In this work we still use the Viterbi training algorithm to optimize the HMM parameters.

Firstly we briefly introduce the Viterbi algorithm, which is used to estimate the optimum hidden state (i.e., to find the most likely class label of each sample data).

To find the optimal state sequence q_1, q_2,⋯,q_t of observation sequence O_1,O_2,⋯,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:

$$\delta_t(i) = \max_{q_1,q_2,\cdots,q_{t-1}} P(q_1 q_2 \cdots q_t = S_i, O_1 O_2 \cdots O_t \,|\, \lambda)\,; \tag{2}$$

Then we obtain

$$\delta_{t+1}(j) = \max_{1\le i\le N}[\delta_t(i) a_{ij}] b_j(O_{t+1})\,; \tag{3}$$

Furthermore, let $\psi_t(j)$ be the optimal state that maximizes $\delta_t(j)$. The procedure of the Viterbi algorithm comprises the following computational steps:

Step 1 - Initialization:

$$\begin{aligned} &\delta_1(i) = \pi_i b_i(O_1), 1 \le i \le N \\ &\psi_1(i) = 0 \end{aligned}\;; \tag{4}$$

Step 2 - Recursion:

$$\begin{aligned} \delta_t(j) &= \max_{1\le i\le N}[\delta_{t-1}(i) a_{ij}] b_j(O_t), 2 \le t \le T, 1 \le j \le N \\ \psi_t(j) &= \arg\max_{1\le i\le N}[\delta_{t-1}(i) a_{ij}], \quad 2 \le t \le T, 1 \le j \le N \end{aligned}\;; \tag{5}$$

Step 3 - Termination:

$$\begin{aligned} P^* &= \max_{1\le i\le N}[\delta_T(i)] \\ q_T^* &= \arg\max_{1\le i\le N}[\delta_T(i)] \end{aligned}\;; \tag{6}$$

Step 4 - Path (state sequence) backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \cdots 1\,. \tag{7}$$

Given the initial HMM parameters, we obtain the class label (finite, discrete state) of each data point using the Viterbi algorithm. Then we can further recalculate the model parameters according to the new class label and Eqs. (3)-(6). This iterative process continues until the algorithm's convergence, that is, the difference of the values of the objective function $P(O,S\,|\lambda)$ between two consecutive iterations is smaller than certain threshold (set as 0.0001 here). The flowchart of the Viterbi training algorithm is shown in Fig. S1.

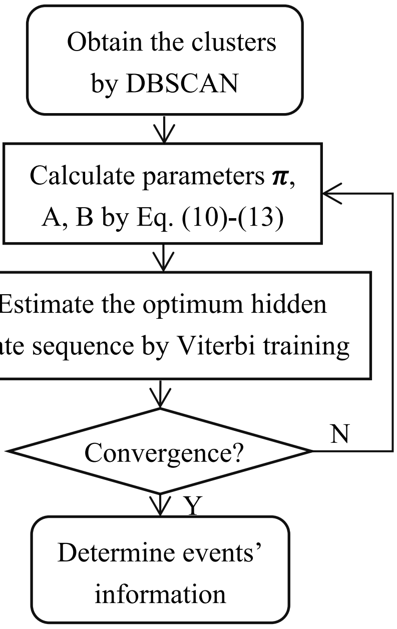
Obtain the clusters by DBSCAN
Calculate parameters $\boldsymbol{\pi}$, A, B by Eq. (10)-(13)
Estimate the optimum hidden state sequence by Viterbi training
Convergence?
N
Y
Determine events' information

Fig. S1 Flowchart of Viterbi training that alternately executes parameter re-estimation and Viterbi algorithm until convergence.

Standard DBSCAN algorithm

In the DBSCAN algorithm, each cluster is assumed to possess a density (defined as the number of neighboring points within a given radius and) that is considerably higher than out of the cluster. This algorithm regards the high-density sample region as the target cluster and the low-density region as the boundary between two clusters.

In the DBSCAN algorithm, we make the following definitions:[12]

Def. 1- ***Eps*-neighborhood of point *p***. For the point *p* in the sample space *D* and the given radius *Eps*, the *Eps*-neighborhood of point ***p*** is the set of points within the radius *Eps*, defined by:

$$N_{Eps}(p) = \{q \in D \mid dist(p,q) \le Eps\}; \tag{8}$$

where *dist*(*p*, *q*) is the Euclidean distance between point *p* and point *q*, the cardinality of $N_{\mathrm{Eps}}(p)$ is called the density of the point *p* given *Eps*. If the density of *p* is larger than the threshold *MinPts*, then *p* is regarded as the core point; otherwise it would be regarded as a boundary point or an outlier.

Def. 2- **Directly density-reachable**. If the two points *p* and *q* in the sample space *D* satisfy the following two conditions, *q* is said to be directly density-reachable from *p*

$$\begin{gathered} q \in N_{Eps}(p) \\ |N_{Eps}(p)| \ge MinPts \end{gathered} \tag{9}$$

Def. 3- **Density-reachable**. Given the sample sequence $p_1,p_2,\cdots,p_n$, and let $p= p_1$, $q= p_n$. If p_{i+1} (i=1,2,$\cdots$,n-1) is directly density-reachable from p_i, we say *q* density-reachable from *p*. We can find that the density-reachability can be regarded as a canonical extension of direct density-reachability. Both relations are transitive but non-symmetric, which means that *q* is density-reachable from *p* does not necessarily imply *p* is density-reachable from *q*. The symmetricity would be satisfied only if both p and q are core points.

Def. 4- **Density-connected**. Given the three points p, q and o in the sample space D, if both p and q are density-reachable from o, we say p and q are density-connected. We can find that density-connectedness satisfies the symmetricity. The DBSCAN clustering is realized by finding the largest set of density-connected samples.

Computational procedure of the modified DBSCAN algorithm

The DBSCAN starts with an arbitrary sample point p. Depending on the parameters *Eps* and *MinPts*, it finds all the density-reachable points from p if p is a core point. Then we add the neighboring points of p to a set (called *seed*), followed by traversing all the points in the seed and performing the same operation for the next point. If p is not a core point, DBSCAN would access the next data point. Although the DBSCAN algorithm does not require the initial knowledge of the number of clusters, the parameters *Eps* and *MinPts* must be pre-set instead. A simple yet effective heuristic approach to setting the two parameters was proposed in previous research.[12] The approach determines the parameters *Eps* and *MinPts* according to the sorted k-dist graph of samples. The parameter *Eps* is the k-dist value of the first inflection point in the sorted k-dist graph. The parameter *MinPts* is the same as k, which usually has little influence on the result. In the following section, we will further examine the effect of parameters using the experimental data.

Step 1. Set the parameters *Eps* and *MinPts*. All the samples are marked as unvisited, and set the cluster number as i=1.

Step 2. Given the two parameters *Eps* and *MinPts* find the unvisited core point p. Set its cluster number as i, and mark p as visited. Then add the maximum and minimum points in the neighborhood of p to the set *seed*.

Step 3. Expand the current cluster: select a point q from *seed*, set its cluster number as i, and mark it as visited, then remove q from the set *seed*.

Step 4. Judge whether or not q is the core point. If so, add the maximum and minimum points in the neighborhood of q to the set *seed*.

Step 5. Repeat **Step 3** and **Step 4** until the set *seed* becomes empty.

Step 6. Class number $i = i + 1$, Repeat Step 2-5 until all the core points are visited.

Step 7. Output clustering results, and mark the unvisited points as outliers.

Based on the clustering results of the DBSCAN algorithm, we can determine the initial model parameters in the following way.

If the class label of the first point in the observation sequence $O_1, O_2, \cdots, O_{\mathrm{T}}$ is i, we can determine

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

The entries in the transition probability matrix A can be determined by[2]

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

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 with S_i and S_j being the cluster number.

We can calculate the mean and variance of each current level by:

$$\mu_i = \frac{\sum_{O_t \in S_i} O_t}{\sum_{O_t \in S_i} 1}$$

$$\sigma_i^2 = \frac{\sum_{O_t \in S_i} (O_t - \mu_i)^2}{\sum_{O_t \in S_i} 1} \tag{12}$$

Then we can determine B from μ_i and σ_i by Gaussian distribution function. After determining the initial values of the HMM parameters, we can use the Viterbi training algorithm to further optimize them.

Optimized the DBSCAN algorithm parameters.

As describe previously, the proposed method only requires the pre-setting of two parameters, Eps and MinPts, involved in the DBSCAN clustering algorithm. To examine the effect of the two parameters using the standard control variable method, we chose the typical event of microRNA21·Probe21. Firstly we fix one parameter *MinPts* to 3 and consider the effect of varying another parameter *Eps* (set as 0.7, 1.5 and 3, respectively). The results under the three different values of *Eps* are compared in Fig. S6a. It is found that if *Eps* is too small (0.7) the noise may be wrongly recognized as current level, but the current level with small amplitude would miss if it is too large (3). Therefore, the optimal value of *Eps* is 1. Then we fixed *Eps* to 1, and varied *MinPts* from 3 to 10 and 30. The results under the three values of the parameter *MinPts* are compared in Fig. S6b. We can see that the manipulation of the parameter *MinPts* hardly influences the recognition results even if it is varied in a wide range between 3 and 30. However, an excessively small *MinPts* would lead to the improper increase of the number of the recognized current levels. The parameters *Eps* and *MinPts* are recommended to take a value in the range [0.5, 3] and larger than 2, respectively. In this work the parameters *Eps* and *MinPts* were set as 1 and 3, respectively.

Computational efficiency of the algorithm

To examine the computational efficiency of the proposed method, we compared the time consumption of the standard and modified DBSCAN algorithm under different sizes of dataset on such a computing hardware/software platform: Intel(R) Core(TM) i5-2450 CPU @2.5GHz, 4G RAM, 64-bit Win7 Pro OS, and Matlab R2013a. We also calculated the total time consumption (DBSCAN + Viterbi training of the HMM) between the standard and modified DBSCAN algorithm. The comparative results are presented in Table S1 and Fig. S7. The results demonstrate that the modified DBSCAN algorithm is at least 3 times faster than the standard DBSCAN. In comparison, the standard DBSCAN algorithm took most of the computational time, while the modified DBSCAN algorithm only account for 7.8% of the total time consumption (the size of the dataset analyzed is 23,378).

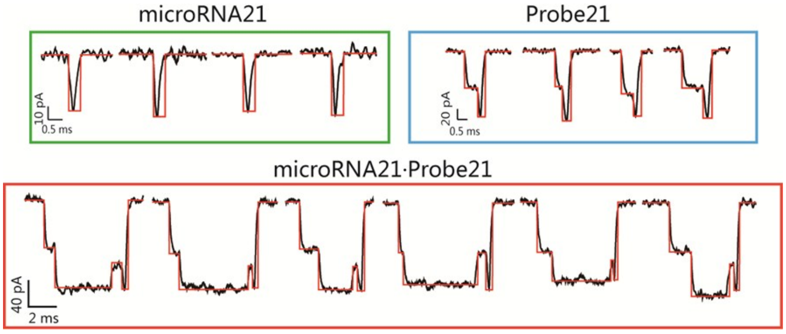

Fig. S2 Typical events acquired by manual analysis of microRNA21 (green box), Probe21 (blue box) and microRNA21·Probe21 (red box). The manual data analysis procure comprises the following steps: Step 1: Load a segment of experimental data; Step 2: Scan the entire signal trace and identify the signal which is identical to the target signal; and Step 3: Finally, evaluate the blockage current amplitude and duration using the cursor in Clampfit.

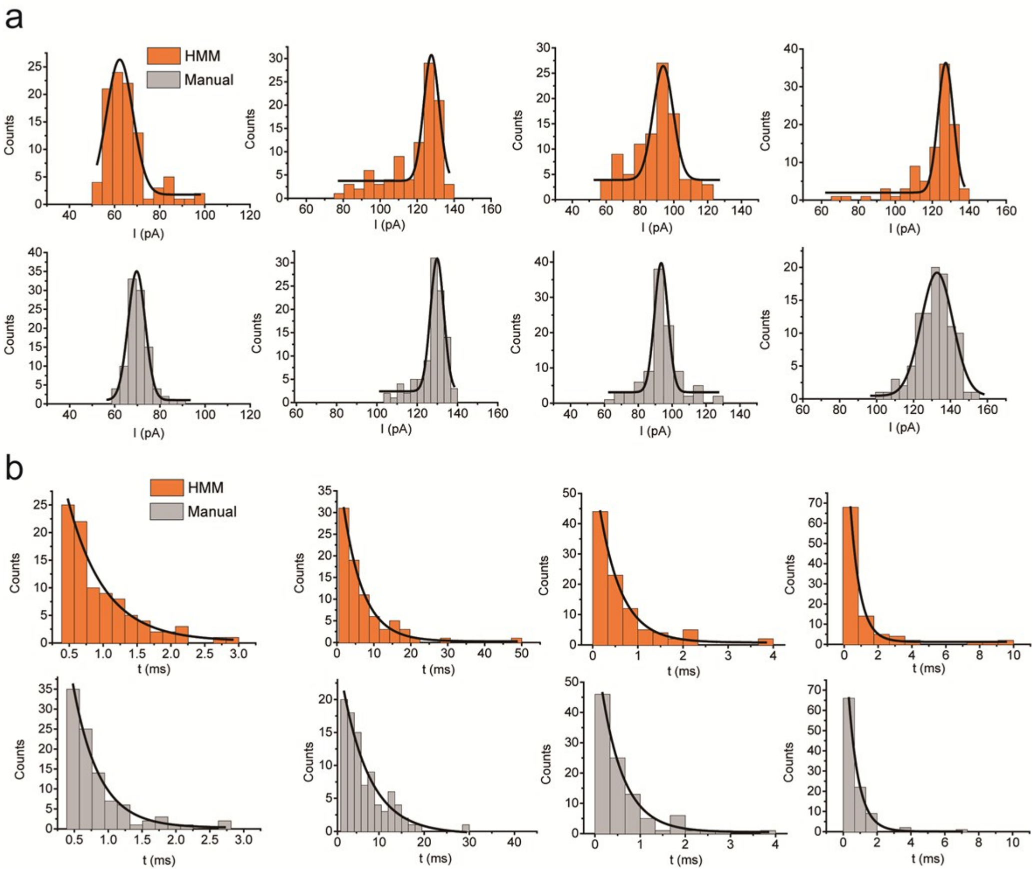

Fig. S3 Current levels' amplitude histogram (a) and duration histogram (b) of miRNA21·Probe21 events acquired by our method (red) and manual analysis (grey). Level 1-4 in order is from left to right.

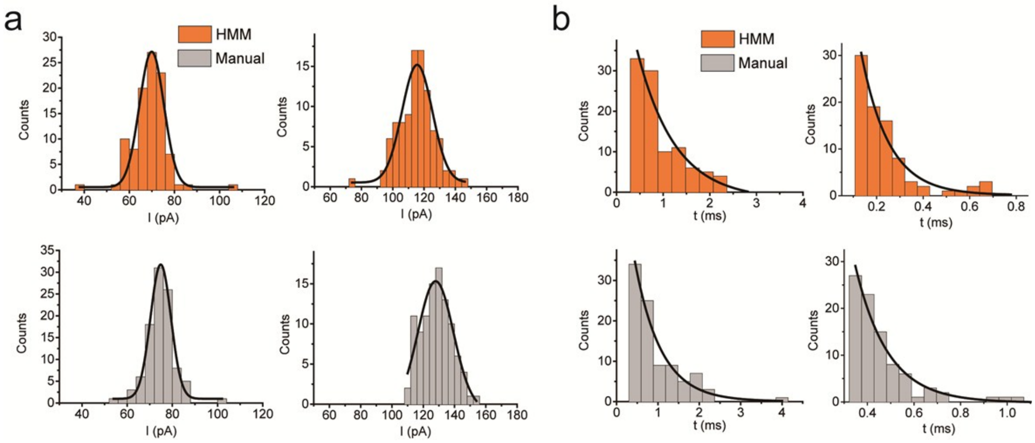

Fig. S4 Current levels' amplitude histogram (a) and duration histogram (b) of Probe21 events acquired by our method (red) and manual analysis (grey). Level 1-4 in order is from left to right.

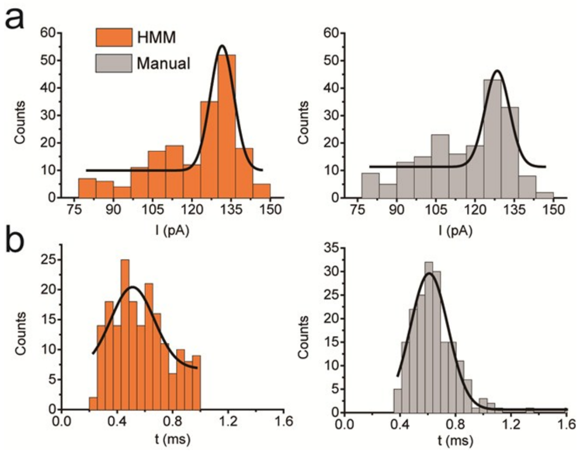

Fig. S5 Current levels' amplitude histogram (a) and duration histogram (b) of microRNA21· events acquired by our method (red) and manual analysis (grey).

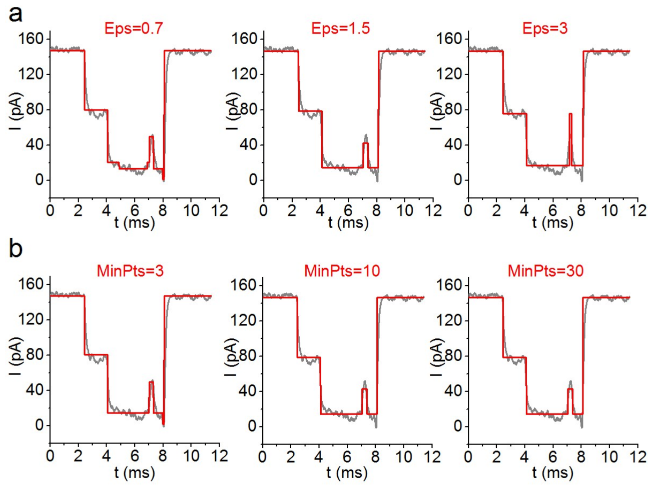

Fig. S6 The results of the DBSCAN algorithm with different parameters: (a) *MinPts*=3; *Eps*=0.7, 1.5, and 3. (b) *Eps*=1; *MinPts*=3, 10, and 30.

Table S1 Comparison of time consumption (s) between the standard and modified DBSCAN algorithm

# of data analyzed	Time consumption of DBSCAN algorithm		Total time consumption of DBSCAN and Viterbi algorithms	
	Standard	Modified	Standard	Modified
761	0.1504	0.0285	0.305	0.182
1974	0.1789	0.0657	0.4296	0.3707
6040	1.305	0.0935	1.8987	0.7304
8457	2.173	0.1169	3.1359	1.1206
13608	5.1679	0.1535	6.6964	1.6886
18653	9.2841	0.2062	11.2618	2.2505
23378	16.6092	0.2679	19.5275	3.4076

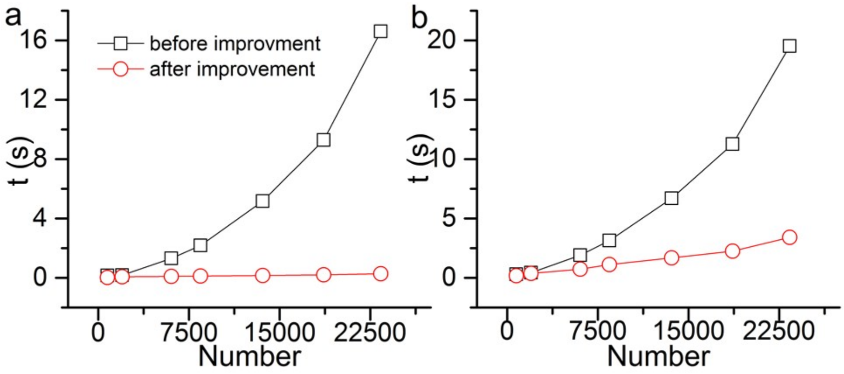

Fig. S7 (a) Time consumption of standard (black) and modified DBSCAN algorithm (red); (b) The total time consumption of the presented method incorporating standard (black) and modified (red) DBSCAN algorithm.

References

(1) Y. L. Ying, J. Zhang, F. N. Meng, C. Cao, X. Yao, I. Willner, Y. T. Long, *Sci. Rep.*, 2013, 3, 1662.

(2) Y. Liu, Y. L. Ying, H. Y. Wang, C. Cao, D. W. Li, W. Q. Zhang, Y. T. Long, *Chem. Commun.*, 2013, 49, 6584-6586.

(3) S. H. Chung, J. B. Moore, L. Xia, L. S. Premkumar, P. W. Gage, *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 1990, **329**, 265-285.

(4) F. Qin, *Biophys. J.*, 2004, **86**, 1488-1501.

(5) L. R. Rabiner, J. G. Wilpon, B. H. Juang, *AT&T Tech. J.*, 1986, **65**, 21-31.

(6) P. A. Devijver, *Pattern Recogn. Lett.*, 1985, **3**, 369-373.

(7) L. R. Welch, *IEEE Information Theory Society Newsletter*, 2003, **53**, 10-13.

(8) G. D. Forney, *Proc. of the IEEE*. 1973, **61**, 268-278.

(9) B. H. Juang, L. R. Rabiner, *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 1990, **38**, 1639-1641.

(10) L. Hu, R. Zanibbi, *Presented at IEEE Intl. Conf. on Document Analysis and Recognition*, Beijing, China, Sept. 18-21, 2011.

(11) J. Zhang, X. Liu,; Y. L. Ying, Z. Gu, F. N. Meng, Y. T. Long, *Nanoscale*, 2017, **9**, 3458-3465.

(12) M. Ester, H. P. Kriegel, J. Sander, X. Xu, *Proc. of the 2nd Intl. Conf. on Knowledge Discovery and Data Mining*, AAAI Press, Portland, 1996, 96, 226-231.