Raw signal segmentation for estimating RNA modifications and structures from Nanopore direct RNA sequencing data

Guangzhao Cheng, Aki Vehtari, and Lu Cheng✉
Department of Computer Science, Aalto University, Finland

Abstract

Estimating RNA modifications from Nanopore direct RNA sequencing data is an important task for the RNA research community. Current computational methods could not provide satisfactory results due to an inaccurate hypothesis in the raw signal segmentation. We present a new method, SegPore, that utilizes a jiggling translocation hypothesis to segment the raw signal. Based on the segmentation results, we demonstrate SegPore's interpretable results and decent performances in m6A and inosine modification estimation, as well as RNA secondary structure estimation. We find that the stop points of the reads take place at the start of estimated stem structures. Our results indicate the SegPore's capability to concurrently estimate multiple modifications at the individual molecule level using the same Nanopore direct RNA sequencing data. Additionally, our study reveals novel perspectives for RNA structure estimation.

Introduction

Nanopore direct RNA sequencing (DRS) measures the current intensity as an RNA molecule translocates through the pore. For Oxford Nanopore Technologies (ONT) pore version R9.4.1, five nucleotides reside in the pore, which is termed as *5mer*[1]. The 5mer blocks certain amounts of ions passing through the pore, the amount of which is measured as the current intensity in a unit of time. During the translocation process, each 5mer stays in the pore for a short period, and the current measurement varies around a baseline level. Due to differences in size, shape, order, and chemical properties of the nucleotides of a 5mer, the current level varies across different 5mers. The basic idea of Nanopore sequencing is to infer the original nucleotides from the currents of different 5mers. As a reference, ONT has provided the mean and standard deviation (std) of the current intensity for each 5mer of RNA based on their training data[2]. Since nucleotides with a larger size could block more ions, e.g., Adenine and Guanine, their current signal levels are lower than Cytosine and Uridine.

RNA modifications play essential roles in different diseases[3-8]. DRS provides a convenient way to probe RNA modifications. There are more than 150 different types of RNA modifications[9-11]. A

✉ Correspondence: lu.cheng.ac@gmail.com.

modification differs from the original nucleotide in its chemical structure, which changes the size, shape and chemical properties of the nucleotide. Consequently, the modified nucleotide exhibits a change in the current signal. In theory, we could estimate all RNA modifications from the raw current signal of DRS. This is much more convenient than probing each modification using different chemical methods. However, this task is challenging from a computational perspective.

The central computational problem of DRS is the segmentation of raw current signal, which is also called the basecalling problem. The raw current signal is split into segments and assigned to corresponding 5mers, which determines the RNA sequence. Current methods such as Guppy and Albacore (https://community.nanoporetech.com) are based on deep neural networks (DNN), which segment the raw signal without interpretation. Consequently, the derived raw signal segments may contain unwanted noises that deteriorate the performance of downstream tasks such as RNA modification estimation.

Another challenge is that an RNA modification may only induce a minor change in the current signal of the corresponding 5mer. This means we can only distinguish 5mers with significant changes in their current signal, i.e., those with a high signal-to-noise ratio. This problem will be mitigated if we could reduce the noise level in the current signal. However, current DNN-based methods do not provide an interpretation of the noise structure, therefore fail to reduce the noise level.

Present RNA modification detection methods can be classified into two groups: comparison based group and single mode group. The comparison based group[12-16] compares the raw signal segments of a modified nucleotide versus an unmodified nucleotide. Due to the unexplainable noises in the raw signal segments, it is difficult to fully differentiate between modified and unmodified nucleotides. The single mode group trains DNN to classify modified and unmodified nucleotides by feeding various features (e.g., baseline, dwell time) of the signal from modification-abundant and modification-sparse datasets[17-20]. Since DNN utilizes multiple features in a complicated way, it is impossible to gain straightforward interpretations of the predicted modification results. Another area for improvement is the availability of large amounts of training data. Both modification-abundant and modification-sparse datasets are needed for each modification separately, which could be very expensive as there are more than 150 RNA modifications.

To address the above challenges, we developed SegPore, which is capable of eliminating noise using a novel signal segmentation algorithm based on an RNA molecule translocation hypothesis and offers the flexibility to estimate various modifications by the customized 5mer parameter table. SegPore consists of (1) a hierarchical hidden Markov model (HHMM) for segmenting the raw current signal of DRS, (2) alignment algorithms for aligning raw current signal segments with 5mers of reference sequence, and (3) a Gaussian mixture model (GMM) for RNA modification estimation. SegPore provides interpretable segmentation results, which facilitates various downstream analyses. We demonstrate its capabilities in the estimation of RNA modifications (m6A and inosine) and RNA secondary structure.

Results

RNA translocation hypothesis

It is challenging to segment the raw current signal due to its complexity. To solve this problem, we need to know how exactly the RNA molecule moves through the pore. In traditional basecalling algorithms such as Guppy and Albacore, we implicitly assume that the RNA molecule is translocated through the pore by the motor protein in a monotonic fashion, i.e., the RNA is pulled through the pore unidirectionally. In the DNN training process of Guppy and Albacore, we try to align the current signal with the reference RNA sequence. The alignment is unidirectional, which is the source of the implicit monotonic translocating assumption.

However, the raw current data suggests that the motor protein translocates RNA back and forth. Fig. 1b shows several example fragments of DRS raw current signal[21], each of which roughly corresponds to three neighboring 5mers. The highlighted spikes have similar current intensities, either as the previous 5mer or the next 5mer. Similar patterns are widely observed across the whole data. This suggests that the RNA molecule may move forward and backward while passing through the pore. This observation can also be supported by previous reports[22, 23], in which the helicase (the motor protein) translocates the DNA strand through the nanopore in a back-and-forth manner.

Based on the reported kinetic model[22], we hypothesize that the RNA is translocated through the pore in a jiggling manner. On average, the motor protein sequentially translocates 5mers on the RNA strand forward, and each 5mer resides in the pore for a short time. During the short period of a single 5mer, the motor protein may swiftly drive the RNA molecule forward and backward by 0.5~1 nucleotide in the translocation process of the current 5mer (Fig. 1a), which makes the measured current intensity occasionally similar to the previous or the next 5mer. When the motor protein does not move the RNA molecule, we assume that the 5mer inside the pore oscillates thermodynamically, generating current intensities around its baseline. Additionally, we assume there is a sharp change in the current intensity between two consecutive 5mers, which serves as their boundary.

We also assume that the raw current signal of a read can be segmented into a series of alternating base and transition blocks (Fig. 1a-b). In the ideal case, a base block corresponds to the base state where the 5mer resides in the pore and jiggles between neighboring 5mers, i.e., the current 5mer can transiently jump to the previous or the next 5mer. A transition block corresponds to the transition state between two consecutive base states where one 5mer translocates to the next 5mer in the pore. The current signal should be relatively flat in the base blocks, while a sharp change is expected in the transition blocks. In practice, the number of transition blocks is generally overestimated since we are not able to distinguish the transitions within a base block and between base blocks. As a result, multiple base blocks may correspond to only one 5mer, which is determined by the SegPore alignment algorithms.

SegPore Workflow

The SegPore workflow (Fig. 2) contains six steps: (1) preprocess fast5 files to pair the raw current signal segments with reference RNA sequence fragments, (2) segment each raw current signal using the hierarchical hidden Markov model (HHMM) into base and transition blocks, (3) align the derived base blocks with the paired RNA sequence, (4) estimate the modification state for each 5mer of the RNA sequence, (5) fit a two-component Gaussian Mixture Model (GMM) for each unique 5mer across different RNA reference sequences, and (6) use results of step 5 to update relevant parameters. Step 3~6 will iterate several times until the estimated parameters stabilize. Detailed descriptions are provided in Methods and Supplementary Note 1-3.

The final outcome of SegPore is the pairing between raw current signal segments and 5mers of the corresponding RNA sequences, which is similar to the output of Nanopolish "eventalign"[24, 25] command. Another key element of SegPore is the 5mer parameter table, which specifies the mean and std of each 5mer in an unmodified or modified state. With the 5mer parameter table, the modification state of each raw current signal segment corresponding to a 5mer can be obtained, i.e., the modification state of each nucleotide of a single RNA molecule.

m6A identification

We demonstrate SegPore's performance on real data segmentation and m6A identification using independent public datasets as training and test data.

We first estimate the 5mer parameter table of m6A modification from the training data using public Nanopore direct RNA sequencing (DRS) data from three wild type samples of HEK293T cells[26]. To get the 5mer parameter table for m6A modification, we concatenate fast5 files of all samples and run the full SegPore workflow. We iterate the parameter estimation process five times to gain a stabilized 5mer parameter table, which contains modification parameters for ten 5mers. Their modification state distribution significantly differs from the unmodified state distribution, with moderate support of read counts and genomics locations.

Next, we perform segmentation and m6A identification on test data, which is the DRS data from wild type mouse embryonic stem cells (mESCs) in a benchmark study[21]. Zhong et al. benchmarked a set of tools with different input requirements. Considering the similarity of the methods and input requirements, we select single mode tools as baselines: Tombo[12], MINES[13], Nanom6A[14], m6Anet[15], and Epinano[16]. We process the test data using standard SegPore workflow with 5mer parameter table estimated from the training data. As a result, we get the alignment of the raw current signal with the reference sequence, modification state of selected 5mers on each read, and modification rate of genomic locations for selected 5mers.

SegPore demonstrates the improved segmentation results compared with Nanopolish. Fig. 3a shows the segmentation results of Nanopolish and SegPore at an example genomic location with m6A modifications. The histogram shows the distribution of the raw signal segment mean

of all reads mapped to the genomic location. Compared with Nanopolish eventalign results, the bimodal distribution is more obvious in SegPore results. We next pooled all reads mapped to modification genomic locations (based on the ground truth) for the classical m6A motif 5mer "GGACT". Fig. 3b illustrates the current signal distribution over all reads mapped to these locations, where prominent peaks are observed in SegPore but not in Nanopolish. The results suggest that SegPore is able to remove a certain degree of noise from the raw current signal, which makes the modification distribution clearer.

SegPore exhibits decent performance on m6A identification on the benchmark data. There are three ground truths based on three different protocols: "MeRIP-seq"[27], "miCLIP"[28], "miCLIP2"[29]. Since the m6A motif frequencies are different in human and mouse, we selected six m6A motifs ("GGACT", "GGACA", "GGACC", "AGACA", "AGACC", "AGACT") that are both abundant in the training and test data (Supplementary Fig. 1). Next, we extract all genomic locations from the mouse reference genome that have the same 5mer as one of the six selected m6A motifs. After that, we only keep genomic locations with ground truth annotations in the benchmark data. In the end, only genomic locations with more than 20 mapped reads are kept. For each genomic location, we calculate the modification rate, which is used to compute the receiver operating characteristic (ROC) curve and precision-recall (PR) curve in Fig. 3c. Prediction results of baseline methods are obtained from Zhang et al.[21] (through personal communications Dr. Luo Guanzheng). The area under the curve of SegPore (AUC) is 72-85% for the ROC curve and 37-40% for the PR curve based on different ground truths (Fig. 3c). SegPore is ranked 2nd in all benchmark comparisons (the higher AUC, the better), which suggests a decent performance of SegPore in m6A identification.

SegPore naturally identifies m6A modifications at the single molecule level. Fig. 3d demonstrates high modification genomic locations (modification rate > 0.1, also supported by ground truth) of an example gene ENSMUSG00000003153, where rows are reads and columns are genomic locations. Biclustering was performed on the heatmap to illustrate the modification patterns, which resulted in 6 clusters of the reads and 3 clusters of the genomic locations. The heatmap suggests that modifications at the 6th, 7th, and 8th genomic locations are specific to cluster 4, 5, 6 of the reads, which correspond to different modification mechanisms. It is also obvious that the majority of reads at the 3rd and 5th genomic locations are modified across other clusters.

Inosine identification

Thanks to the flexibility of the 5mer parameter table, SegPore is proficient in estimating various RNA modifications. We next test the performance of SegPore on inosine identification, which is another RNA modification of adenine[30-33]. The *in virto* transcription (IVT) data and H9 human embryonic stem cells (hESCs) data in a public dataset[33] were used for this task. The 5mer parameter table of inosine was obtained from IVT data, and the H9 data was for testing.

First, we explain the training process. The IVT data contained eight pure-G samples and eight pure-I samples, which were merged into one pure-G and one pure-I sample, respectively. The

pure-G sample was an unmodified sample where A, C, G, U nucleotides were used in the synthesis of a designed RNA molecule. The designed sequence contains 81 different XXGXX 5mers, where X is A, C or U/T. The pure-I sample was a modified sample where A, C, I (inosine), U were used for synthesizing the same RNA molecule. The “G” nucleotides in the RNA sequence could either be replaced by A or I nucleotides, which provided data to infer distributions of 5mers in the form of XXA/IXX. We run standard SegPore workflow on the pure-G sample and pure-I sample to get a stabilized 5mer parameter table, which contains modification states for 61 selected 5mers (Supplementary Fig. 2a). Note that the mean of XXGXX and XXAXX are fixed, whose values are specified by the 5mer parameter table of ONT.

The derived 5mer parameter table nicely depicts the selected 5mer distributions. Fig. 4a shows three example patterns of 5mer distributions, which are the densities derived from raw signal segments aligned to the selected 5mers on the reference sequence. We obtain the XXGXX distributions from the pure-G sample and XXA/IXX distributions from the pure-I sample. For the pure-G sample, we expect only one peak for XXGXX, while we expect two peaks for XXAXX and XXIXX in the pure-I sample. There are three general patterns: (1) G is replaced by A or I in the pure-I sample, but we cannot differentiate the peaks of XXAXX and XXIXX (top panel) (2) G is replaced by A or I, and the peaks of A and I can be clearly identified (center panel) (3) G is only replaced by I and the peak of I is significantly different to the ONT reference A peak (bottom panel). The selected 5mers belong to the second and third patterns (Supplementary Fig. 2a). Supplementary Fig. 2b shows that the mean of XXIXX is generally between that of XXAXX and XXGXX, which agrees with the findings in the original publication[33].

We test the performance of SegPore on H9 DRS data, which consists of ten wild type samples. We used the 5mer table estimated from the training data and ran the standard SegPore workflow on H9 data, which provides the modification rate of each genomic location that matches the selected 5mers. Due to data availability, we only find the ground truth of locations on Chromosome 3 and 11 from the original publication[33]. Out of these locations, 37 inosine modification sites match our selected 5mers, which are used as the positive control, and 241 unmodified sites match our selected 5mers, which are used as the negative control. Fig. 4b shows the current signal distribution of one selected 5mer CAACA by pooling all relevant sites on positive control, which exhibits two clear peaks for CAACA and CAICA. This suggests a high consistency of 5mer distributions between the IVT data and the H9 data. Fig. 4c shows the modification rate distribution in the positive and negative controls. It can be seen that the modification rates of positive control are higher than that of negative control on average (t-test, p-value=2.22e-11).

RNA structure estimation

RNA structure can be probed by sequencing based approaches such as SHAPE-MaP[34] and nanoSHAPE[35]. Their idea is to treat the RNA molecules with chemical adducts, which are only attached to single stranded parts but not double stranded parts of RNA molecules. The chemically treated RNA is then subjected to short-read sequencing (SHAPE-Map) or Nanopore sequencing (nanoSHAPE). In short-read sequencing, a nucleotide attached by an adduct turns

out to be a mutation, while it is recognized as a modification in Nanopore sequencing. The mutation rate or modification rate is then transformed to a reactivity score that is utilized by the RNAfold software[36] to generate RNA structure.

SegPore can be used to identify the modifications and predict RNA structure from Nanopore sequencing data. As an illustration, we analyze the public Nanopore sequencing data of pri-miR-17~92 RNA in the nanoSHAPE paper[35], for which the ground truth structure (based on SHAPE-MaP) is available. In this dataset, we choose Nanopore sequencing data of two samples: one *in vitro* transcribed (IVT) sample with no modifications and one adduct treated sample with modifications. Following standard SegPore workflow, we first segment raw current signals of reads in both samples. Since all reads are from the same reference sequence, we directly align the derived signal segments with 5mers of the pri-miR-17~92 reference sequence. Then, for each location (5mer) of the reference sequence, we fit a two-component Gaussian mixture model to all signal segments aligned to that location in both samples, with one component representing the unmodified state and the other representing the modified state. Next, we estimate the modification rate and reactivity score for each location of the reference sequence based on the fitted GMM parameters. In the end, pri-miR-17~92 RNA sequence and the reactivity scores are fed into RNAfold software[36] to generate the final RNA structure. Details of the method are provided in Supplementary Note 3.

The RNA structure given by SegPore nicely agrees with the ground truth (SHAPE-MaP), as shown in Fig. 5b. The conserved six hairpin structures, each marked by a dashed line in the center, are almost identical between SegPore and SHAPE-MaP predictions. Six example sites in the loop part from each hairpin structure are selected to illustrate the modifications in Fig. 5a, where a large component of modification (highlighted in red) is observed in the modified sample. This result is consistent with the expected outcome of the SHAPE protocol as the loops are single stranded, where adducts are attached and recognized as modifications. The reactivity scores of SHAPE-MaP and SegPore are similar (Fig. 5c), and peaks are located at similar sites. It can be seen that peak regions in Fig. 5c generally correspond to single stranded regions with no base pairing in Fig. 5b.

After aligning the raw signal with the reference sequence, large amounts of reads in both unmodified (86.3%) and modified samples (95.7%) do not cover the full-length reference sequence. We are curious about the cause and examine the distribution of the alignment end locations on the reference sequence. Fig. 5f shows that the distributions are very similar between the unmodified and modified samples, which precludes the adduct modifications from being the cause. Comparing the peak regions in Fig. 5f with the RNA structure in Fig. 5d and Fig. 5e, we find that the peak regions are generally located right before stem structures, i.e., a stretch of paired bases. Note that the reverse transcription starts from 3’ to 5’, it is obvious that the peak regions are right next to the downstream stem structures, the start locations of which are highlighted by crossing signs in Fig. 5d.

The above finding suggests that peaks are associated with reverse transcription in the Nanopore sequencing library preparation. Here, we hypothesize that the reverse transcriptase

enzyme loses its momentum when it hits stem structures highlighted by crossings in Fig. 5d, as it takes lots of energy to unwind the stem structures. When the reverse transcriptase enzyme stalls on the RNA molecule, we hypothesize that the RNA molecule fractures near the stalled locations due to the thermodynamic movements of the reverse transcriptase enzyme. Therefore, a partial read corresponds to a fractured RNA molecule in the reverse transcription.

Discussion

The key computational problem in DRS is how to segment the raw current signal. We developed a segmentation algorithm that utilized the jiggling property in the physical process of DRS and demonstrated that the segmentation led to cleaner current signals in m6A, inosine identification, as well as RNA structure estimation. We believe that cleaner current signals will be beneficial for various downstream tasks. However, some open questions remain to be addressed in the future. In SegPore, we assume a drastic change between two consecutive 5mers, which may hold for 5mers with large difference in their current baselines but may not hold for those with small difference. Another question is the physical meaning of derived base blocks. In the ideal case, one base block corresponds to one 5mer, while multiple base blocks are aligned with one 5mer in the real case. One guess is that the HHMM may partition the current signal of one 5mer into several base blocks, during which the 5mer may oscillate between different sub-states, i.e., each sub-state of the same 5mer has different baselines.

Different from DNN-based methods, SegPore offers great interpretability to the estimation of RNA modifications, which makes it applicable to different modification states. The SegPore's algorithm codes the current intensity levels for different 5mers in a parameter table, where unmodified and modified 5mers are modeled using Gaussian distributions. Given 5mer parameter tables of different RNA modifications, e.g., m6A, inosine, pseudouridine[37-39], etc., we can estimate different modifications from the same DRS data. Since we directly model the current level of 5mers with RNA modifications, SegPore naturally provides the modification states on the single RNA molecule level. This capability will be significantly powerful for studying RNA modifications in various disease contexts. However, the amount of 5mers with significant changes in their modification states is relatively small. We may need larger training data to improve the accuracy and expand 5mers to 7mers or 9mers to consider more context information, on which significant baseline change might be observed on more kmers.

In RNA structure experiment, we also found there was an association between the end points of reads and the stem structures of the RNA molecule. We hypothesize that the RNA molecule fractures are near the stem structures. If stronger stem structures are associated with a larger proportion of partial reads, we may use this information to probe the binding energy of the stems and incorporate it in the RNA structure estimation, which is a unique angle provided by DRS. SegPore also increased the mapping rate of full-length RNA sequences. In our experiment, SegPore identified 21% of reads to be full-length, while Nanopolish identified 5%.

Computation speed is another concern in handling the fast5 files. We also implemented a GPU-accelerated inference algorithm in SegPore, which has a 10~20 fold speedup compared with CPU-based implementations.

Methods

SegPore workflow

Preprocessing

We first perform basecalling of the input fast5 file using Guppy. Then we map the basecalled sequence to the reference sequences using Minimap2[40]. After that, we obtain the paired raw current signal segments and corresponding fragments of the reference sequence using Nanopolish eventalign[24, 25]. Meanwhile, we obtain the raw current signal segments corresponding to the polyA tail. Finally, we standardize the raw current signal of each read based on its polyA tail, such that the mean and std of the polyA tail are the same across different reads.

Signal Segmentation via hierarchical Hidden Markov model

The RNA translocation hypothesis naturally leads to a hierarchical hidden Markov model (HHMM) for segmenting the raw current signal. As shown in Fig. 2b, our HHMM has two layers. The outer HMM divides the raw current signal into alternating base and transition blocks, as indicated by hidden states “B” and “T” in Fig. 2b. The inner HMM models a single base block and a linear model is used for each transition block. The inner HMM has four hidden states: “prev”, “next”, “curr”, “noise”. The “prev”, “next”, “curr” states refer to previous, next and current 5mer in the pore, while “noise” refers to random noise. Each raw current measurement is then emitted from one of these hidden states. The linear model with a large absolute slope is used to model the sharp changes in the transition block.

Given the raw current signal $\boldsymbol{y}$ of a read, we denote the hidden states of the outer hidden HMM by $\boldsymbol{g}$. $\boldsymbol{y}$ and $\boldsymbol{g}$ are divided into $2K+1$ blocks $\boldsymbol{c}$, where $\boldsymbol{y}^{(k)}$, $\boldsymbol{g}^{(k)}$ correspond to k th block and $\boldsymbol{c}=(c_1,c_2,...,c_k,...,c_{2K+1}), c_k \in \{"B","T"\}$. Note that blocks with odd index are base blocks, i.e. $k=1,3,5,...,2K+1$, whereas blocks with even index are transition blocks. The likelihood of HHMM is given by

$$p(\boldsymbol{y},\boldsymbol{g}) = p(\boldsymbol{y}\mid\boldsymbol{g})p(\boldsymbol{g}) \tag{1}$$

$$= p(\boldsymbol{y}\mid\boldsymbol{g})\{\pi^{outer}_{g_1}\prod_{i=2}^{N}T^{outer}_{g_{i-1}g_i}\} \tag{2}$$

$$= \{\prod_{i=1}^{N}p(y_i\mid g_i)\}\{\pi^{outer}_{g_1}\prod_{i=2}^{N}T^{outer}_{g_{i-1}g_i}\} \tag{3}$$

$$= \{\prod_{k=0}^{K}p(\boldsymbol{y}^{(2k+1)}\mid c_{2k+1}="B")\}\{\prod_{k=1}^{K}p(\boldsymbol{y}^{(2k)}\mid c_{2k}="T")\}\{\pi^{outer}_{g_1}\prod_{i=2}^{N}T^{outer}_{g_{i-1}g_i}\} \tag{4}$$

where $T^{outer}_{g_{i-1}g_i}$ is the transition matrix of the outer HMM and $\pi^{outer}_{g_1}$ is the probability for the first hidden state. It is obviously seen that the left side of the Eq. 2 are emission probabilities, and the right side are the transition probabilities. It is not possible to directly compute the emission probabilities of the outer HMM (Eq. 3) since there exist dependencies for the current signal measurements within a base or transition block. Therefore, we use the inner HMM and linear model (Eq. 4) to handle the dependencies and approximate emission probabilities.

The inner HMM models the transitions between the hidden states “prev”, “next”, “curr”, and “noise”. For “prev”, “next”, “curr” states, we use the Gaussian distribution as their emission distribution. Uniform distribution is used as the emission distribution for the “noise” state. Given these specifications, we are able to calculate the marginal likelihood of the inner HMM using the Forward-Backward algorithm, which is used to approximate the emission probabilities of the base blocks in the outer HMM. Similarly, we can approximate the emission probabilities of the transition blocks using the likelihood of the standard linear model.

For any given $\boldsymbol{g}$, we can calculate the joint likelihood (Eq. 1). Therefore, we enumerate different configurations of $\boldsymbol{g}$ and choose the one with the highest likelihood. Detailed model description is provided in Supplementary Note 1.

The parameter inference is challenging given the massive data size of fast5 files, which are generally on the level of terabytes (TB). We developed a GPU-based inference algorithm to accelerate the inference process. A detailed description of the GPU-accelerated inference algorithm is provided in Supplementary Note 2.

In the end, we segment the raw current signal of a read into alternating base and transition blocks, where one or multiple base blocks may correspond to only one 5mer. For each base block, it has a mean and std parameter (Gaussian distribution). The mean values of the base blocks are used as input for the downstream alignment tasks.

Alignment of raw signal segment with reference sequence

After segmenting the raw current signal of a read into base and transition blocks using HHMM, we align the means of base blocks with the 5mer list of the reference sequence.

The alignment has three different matching cases. The first case is that one base block matches with one 5mer, which means the base block follows the Gaussian distribution of the given 5mer. Note that the 5mer might have two states: unmodified and modified. The corresponding Gaussian parameters can be found in the 5mer parameter table. The second case is the one base block matches with an indel “-”, which means there is an inserted nucleotide in the read. The third case is that an indel (0.0) matches with a 5mer, which means there is a deleted nucleotide in the read.

Our score function in the alignment models the matching cases as follows. For the first case, we calculate the probability of the base block mean sampled from the unmodified 5mer Gaussian

distribution, as well as the modified 5mer Gaussian distribution. The larger probability is used as the match score in this case. For the second or third case, we treat it as noise and use a fixed uniform distribution to calculate the match score.

In our alignment, another significant difference with the classical global alignment algorithm is that one or multiple base blocks could be aligned with one 5mer. Given the base block means $\boldsymbol{\mu} = (\mu_1, \mu_2, ..., \mu_i, ..., \mu_m)$ and 5mer list $\boldsymbol{s} = (s_1, s_2, ..., s_j, ..., s_n)$, we define $(m+1) \times (n+1)$ the score matrix as $\boldsymbol{M}$ (Fig. 2c). The first row and column of $\boldsymbol{M}$ are reserved for indels "0.0" and "-", which represent indels of the base block and 5mer, respectively. We denote the score function by f. The recursion formula of the dynamic programming alignment algorithm is given by

$$\boldsymbol{M}(i,j) = max \begin{cases} \boldsymbol{M}(i-1, j-1) + f(\mu_i, s_j) \\ \boldsymbol{M}(i, j-1) + f(0.0, s_j) \\ \boldsymbol{M}(i-1, j) + f(\mu_i, "-") \\ \boldsymbol{M}(i-1, j) + f(\mu_i, s_j) \end{cases} . \quad (5)$$

It can be seen that we can still align μ_i with s_j after we have aligned μ_{i-1} with s_j, which fulfills the special consideration that one or multiple base blocks might be aligned with one 5mer.

There are two different types of alignment algorithms given the score matrix $\boldsymbol{M}$: the full alignment algorithm and the partial alignment algorithm (Fig. 2c). The full alignment algorithm tries to align the full list of base block means with the full list of 5mer list, which is similar to the classical global alignment algorithm. This is implemented by tracing back from the $(m+1, n+1)$ position of the score matrix. The partial alignment algorithm tries to align the full list of the base blocks with the first part of the 5mer list. It differs from the full alignment algorithm in two aspects: (1) no indel is allowed in both the base block means and the 5mer list, and (2) trace back the maximum value of the last row of the score matrix $\boldsymbol{M}$. A detailed description of the full and partial alignment algorithm is provided in Supplementary Note 1.

The output of the alignment algorithm is the pairing between the base blocks and 5mer list of reference sequences for each read (Fig. 2c).

Modification estimation

After obtaining the alignment results of base blocks and 5mer list of the reference sequence, we estimate the modification state of each 5mer using the 5mer parameter table. If the 5mer is in the modification state, the probability of the base block mean under the modified 5mer Gaussian distribution should be higher than that calculated using unmodified 5mer parameters, and vice versa. Since one 5mer may be aligned with multiple base blocks, we merge all the aligned base blocks and take the weighted mean as the single base block mean aligned with the given 5mer, which provides us the modification state of each site of a read. We then pool all reads mapped to the same genomic location on the reference sequence to get the modification rate of the genomic location, which is the proportion of reads in modification state. Detailed description of the modification state estimation is provided in Supplementary Note 1.

GMM for 5mer parameter table re-estimation

To gain better alignment results and more accurate modification estimation, we use GMM to fine-tune the 5mer parameter table iteratively. As shown in Fig. 2a, the row of the 5mer parameter table are 5mers and the columns are the mean and std of the unmodified and modified states. We denote a 5mer by s and its relevant parameters by $\mu_{s,un}$, $\delta_{s,un}$, $\mu_{s,mod}$, $\delta_{s,mod}$.

Given the alignment results of all reads, we extract all base block means that are aligned to the same 5mer s on different reads and across different genomic locations with high modification rates. Next, we fit a two-component GMM to the collected base blocks corresponding to 5mer s, with the first component mean fixed to $\mu_{s,un}$. From the GMM, we have the updated $\delta_{s,un}$, $\omega_{s,un}$, $\mu_{s,mod}$, $\delta_{s,mod}$, and $\omega_{s,mod}$, where $\omega_{s,un}$, $\omega_{s,mod}$ are the weights for unmodified and modified components. Then, we manually update the 5mer parameter table based on some heuristics such that the modified 5mer distribution is significantly different from that of unmodified 5mer distribution. Detailed description of the GMM re-estimation process is provided in Supplementary Note 1.

The re-estimation of 5mer parameter table is only performed on the training data. We initialize the 5mer parameter table using the 5mer parameter table provided by ONT. Every time after we gain the updated 5mer parameter table, we run the SegPore workflow from the alignment again. The process is repeated 3 to 5 times in general, after which the 5mer parameter table is stabilized. After that, the 5mer parameter table is fixed. For testing data, we estimate the RNA modification states using the fixed 5mer parameter table.

m6A benchmark

Fasta files of the training dataset (3 HEK293T WT samples) were downloaded from European Nucleotide Archive (ENA) database under accession number PRJEB40872, consisting of the following files: HEK293T-WT-rep1.tar.gz, HEK293T-WT-rep2.tar.gz, and HEK293T-WT-rep3.tar.gz. The corresponding fast5 files and reference sequence (Homo_sapiens.GRCh38.cdna.ncrna_wtChrIs_modified.fa) were downloaded from https://doi.org/10.5281/zenodo.4587661. The ground truth was downloaded from Supplementary Data 1 of Pratanwanich, P.N. et al.[26]. Fast5 files of the test dataset (mESC WT samples, mESCs_Mettl3_WT_fast5.tar.gz) were downloaded from NCBI Sequence Read Archive (SRA) database under accession number SRP166020.

During training, the 5mer parameter table was initialized using ONT. Standard SegPore workflow is performed on the training data, where the full alignment algorithm is used. The 5mer parameter table estimation was iterated five times. During the training, reads were first mapped to cDNA, then converted to genomic locations on the reference genome using Ensembl GTF file (GRCh38, v9), after that the same 5mer at different genomic locations were pooled together. We select 5mers with significant modifications if its read coverage is greater than 1,500 and the

distance between two components means in the GMM is greater than 5. The modification parameters were specified for ten significant 5mers, as illustrated in Supplementary Fig. 1a.

With the estimated 5mer parameter table from the training data, we ran SegPore workflow on the test data. Transcript sequences of GENCODE release version M18 were used as the reference sequence for mapping, where the GTF file (gencode.vM18.chr_patch_hapl_scaff.annotation.gtf) downloaded from GENCODE was used to convert transcript locations to genomic locations. Note that we do not estimate 5mer parameter table for test data, and the modification states for each read are estimated only once. Due to the difference between human and mouse, only six out of ten selected 5mer have m6A annotations in the ground truth of the test data (Supplementary Fig. 1c). For a genomic location to be considered as a m6A modification site, we require that it must correspond to one of the six common 5mers and the read coverage must be greater than 20.

The modification rate of each genomic location was used to derive the ROC and PR curves in the benchmark study. The ground truth, performances of other methods (Tombo v1.5.1, Nanom6A v2.0, m6Anet v1.0, and Epinano v1.2.0) were obtained through personal communications with Dr. Luo Guanzheng, who is the corresponding author of the referenced benchmark study[21].

Inosine identification experiments

Raw nanopore sequencing data of *in virto* transcription data (training data) and H9 human embryonic stem cells (test data) were downloaded from NCBI Sequence Read Archive under accession number SRP363295. The training data consists of eight pure-G samples (Accessions: SRX18177003, SRX14536452, SRX14536451, SRX14536450, SRX14536449, SRX14536448, SRX14536447, and SRX14536446), eight pure-I samples (Accessions: SRX18176999, SRX14535372, SRX14535371, SRX14535370, SRX14535369, SRX14535368, SRX14535367, and SRX14535366), and three reference sequences for synthetic RNA can be download from Supplementary Table 5 of the original publication[33]. The test data consists of ten H9 WT embryo samples (Accessions: SRX14436756, SRX14436755, SRX14436754, SRX14436753, SRX14436752, SRX14448128, SRX14448125, SRX14448129, SRX14604385, and SRX14604391). The reference sequence for the test data is the human GRCh37 cDNA sequences downloaded from the Ensembl database.

Standard SegPore workflow was performed on the training data, where four iterations were used to derive the 5mer parameter table. Considering that there is only one transcript per sample in the IVT data, the partial alignment algorithm is used in the training. The designed reference sequences of the IVT data contain 81 different 5mers in the form of XXGXX, where X is A, C or U/T. We manually classify the 81 5mers into three patterns (Supplementary Fig. 2a), which contain 20, 23, 38 5mers, respectively. For Pattern 2, a two-component GMM is fitted to the data with the unmodified component mean fixed. For Pattern 3, a single Gaussian distribution is fitted to the data with the constraint that the std should not exceed 3.0. The final

5mer parameter table after training contains modification parameters for 61 5mers (Pattern 2 and 3).

The derived 5mer parameter table was then used to identify inosine from the test data (H9 data). Ensembl GTF file (GRCh37, v87) was used to convert transcript locations to genomic locations. For a genomic location to be considered an inosine modification site, its read coverage must be larger than 10. Due to the availability of data, we can only download the ground truth on Chromosome 3 and 11 (test_H9_regression_allannotatedsites_0to1.RData) for the test data from the code repository of the original publication[33], which contains the modification rates of 4,934 genomic locations. From these genomic locations, we selected sites with a modification rate greater than 0.1 as the positive control, and sites with a modification rate equal to 0 as the negative control. As a result, 1,007 sites were retained from the ground truth, consisting of 129 positive sites (37 unique 5mers) and 878 negative sites. Out of the 37 unique 5mers in the positive control of the ground truth, there are 18 overlapping 5mers with the 61 selected 5mers (Pattern 2 and 3) in the training. The sites that match the 18 overlapping 5mers were kept, after that we retained sites with read coverage larger than 10, resulting in 37 positive sites and 241 negative sites. SegPore estimated modification rates of these sites were extracted to generate Fig. 4c.

Code Availability

The source code is hosted on GitHub (https://github.com/guangzhaocs/SegPore).

Acknowledgements

We would like to thank Prof. Zhijie Tan from Wuhan University for a useful discussion about the molecule dynamics of Nanopore sequencing, Dr. Dan Zhang from Sichuang University for helpful tutorials about traditional Nanopore analysis workflows. We would like to thank Research Council of Finland grants (NO. 335858, 358086) to GC and LC. GC and LC acknowledge the computational resources provided by the Aalto Science-IT project.

Author information

Authors and Affiliations

Department of Computer Science, Aalto University, Finland
Guangzhao Cheng, Aki Vehtari, and Lu Cheng*

Contributions

GC developed the methods and performed the analyses. AV provided advice about statistical modelling and manuscript writing. LC conceptualized and supervised the project. GC and LC co-implemented the SegPore workflow, co-wrote the manuscript. All authors read and approved the manuscript.

Corresponding author
Correspondence to Lu Cheng (lu.cheng.ac@gmail.com).

Ethics declarations

Competing interests
The authors declare no competing interests.

References

1. Jain, M. et al. Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nat Biotechnol* **36**, 338-345 (2018).
2. ONT. kmer_models. https://github.com/nanoporetech/kmer_models (2023).
3. Barbieri, I. & Kouzarides, T. Role of RNA modifications in cancer. *Nat Rev Cancer* **20**, 303-322 (2020).
4. Cui, L. et al. RNA modifications: importance in immune cell biology and related diseases. *Signal Transduct Target Ther* **7**, 334 (2022).
5. Mathiyalagan, P. et al. FTO-Dependent N(6)-Methyladenosine Regulates Cardiac Function During Remodeling and Repair. *Circulation* **139**, 518-532 (2019).
6. Li, Z. et al. FTO Plays an Oncogenic Role in Acute Myeloid Leukemia as a N(6)-Methyladenosine RNA Demethylase. *Cancer Cell* **31**, 127-141 (2017).
7. Su, R. et al. R-2HG Exhibits Anti-tumor Activity by Targeting FTO/m(6)A/MYC/CEBPA Signaling. *Cell* **172**, 90-105 e123 (2018).
8. Deng, X., Su, R., Weng, H., Huang, H., Li, Z. & Chen, J. RNA N(6)-methyladenosine modification in cancers: current status and perspectives. *Cell Res* **28**, 507-517 (2018).
9. Boccaletto, P. et al. MODOMICS: a database of RNA modification pathways. 2021 update. *Nucleic Acids Res* **50**, D231-D235 (2022).
10. Zimna, M., Dolata, J., Szweykowska-Kulinska, Z. & Jarmolowski, A. The expanding role of RNA modifications in plant RNA polymerase II transcripts: highlights and perspectives. *J Exp Bot* **74**, 3975-3986 (2023).
11. Ohira, T. & Suzuki, T. Transfer RNA modifications and cellular thermotolerance. *Mol Cell* **84**, 94-106 (2024).
12. Stoiber, M. et al. De novo Identification of DNA Modifications Enabled by Genome-Guided Nanopore Signal Processing. *biorxiv* (2017).
13. Lorenz, D.A., Sathe, S., Einstein, J.M. & Yeo, G.W. Direct RNA sequencing enables m(6)A detection in endogenous transcript isoforms at base-specific resolution. *RNA* **26**, 19-28 (2020).
14. Gao, Y. et al. Quantitative profiling of N(6)-methyladenosine at single-base resolution in stem-differentiating xylem of Populus trichocarpa using Nanopore direct RNA sequencing. *Genome Biol* **22**, 22 (2021).
15. Hendra, C., Pratanwanich, P.N., Wan, Y.K., Goh, W.S.S., Thiery, A. & Goke, J. Detection of m6A from direct RNA sequencing using a multiple instance learning framework. *Nat Methods* **19**, 1590-1598 (2022).
16. Liu, H. et al. Accurate detection of m(6)A RNA modifications in native RNA sequences. *Nat Commun* **10**, 4079 (2019).
17. Wick, R.R., Judd, L.M. & Holt, K.E. Performance of neural network basecalling tools for Oxford Nanopore sequencing. *Genome Biol* **20**, 129 (2019).
18. Huang, N., Nie, F., Ni, P., Luo, F. & Wang, J. SACall: A Neural Network Basecaller for Oxford Nanopore Sequencing Data Based on Self-Attention Mechanism. *IEEE/ACM Trans Comput Biol Bioinform* **19**, 614-623 (2022).
19. Liu, Q., Fang, L., Yu, G., Wang, D., Xiao, C.L. & Wang, K. Detection of DNA base modifications by deep recurrent neural network on Oxford Nanopore sequencing data. *Nat Commun* **10**, 2449 (2019).
20. Senanayake, A., Gamaarachchi, H., Herath, D. & Ragel, R. DeepSelectNet: deep neural network based selective sequencing for oxford nanopore sequencing. *BMC Bioinformatics* **24**, 31 (2023).
21. Zhong, Z.D. et al. Systematic comparison of tools used for m(6)A mapping from nanopore direct RNA sequencing. *Nat Commun* **14**, 1906 (2023).

22. Craig, J.M. et al. Revealing dynamics of helicase translocation on single-stranded DNA using high-resolution nanopore tweezers. *Proc Natl Acad Sci U S A* **114**, 11932-11937 (2017).
23. Caldwell, C.C. & Spies, M. Helicase SPRNTing through the nanopore. *Proc Natl Acad Sci U S A* **114**, 11809-11811 (2017).
24. Loman, N.J., Quick, J. & Simpson, J.T. A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nat Methods* **12**, 733-735 (2015).
25. Simpson, J.T., Workman, R.E., Zuzarte, P.C., David, M., Dursi, L.J. & Timp, W. Detecting DNA cytosine methylation using nanopore sequencing. *Nat Methods* **14**, 407-410 (2017).
26. Pratanwanich, P.N. et al. Identification of differential RNA modifications from nanopore direct RNA sequencing with xPore. *Nat Biotechnol* **39**, 1394-1402 (2021).
27. Zhang, Z. et al. Systematic calibration of epitranscriptomic maps using a synthetic modification-free RNA library. *Nat Methods* **18**, 1213-1222 (2021).
28. Ke, S. et al. m(6)A mRNA modifications are deposited in nascent pre-mRNA and are not required for splicing but do specify cytoplasmic turnover. *Genes Dev* **31**, 990-1006 (2017).
29. Kortel, N. et al. Deep and accurate detection of m6A RNA modifications using miCLIP2 and m6Aboost machine learning. *Nucleic Acids Res* **49**, e92 (2021).
30. Slotkin, W. & Nishikura, K. Adenosine-to-inosine RNA editing and human disease. *Genome Med* **5**, 105 (2013).
31. Eisenberg, E. & Levanon, E.Y. A-to-I RNA editing - immune protector and transcriptome diversifier. *Nat Rev Genet* **19**, 473-490 (2018).
32. Nishikura, K. A-to-I editing of coding and non-coding RNAs by ADARs. *Nat Rev Mol Cell Biol* **17**, 83-96 (2016).
33. Nguyen, T.A. et al. Direct identification of A-to-I editing sites with nanopore native RNA sequencing. *Nat Methods* **19**, 833-844 (2022).
34. Siegfried, N.A., Busan, S., Rice, G.M., Nelson, J.A. & Weeks, K.M. RNA motif discovery by SHAPE and mutational profiling (SHAPE-MaP). *Nat Methods* **11**, 959-965 (2014).
35. Stephenson, W., Razaghi, R., Busan, S., Weeks, K.M., Timp, W. & Smibert, P. Direct detection of RNA modifications and structure using single-molecule nanopore sequencing. *Cell Genom* **2** (2022).
36. Lorenz, R. et al. ViennaRNA Package 2.0. *Algorithms Mol Biol* **6**, 26 (2011).
37. Rodell, R., Robalin, N. & Martinez, N.M. Why U matters: detection and functions of pseudouridine modifications in mRNAs. *Trends Biochem Sci* **49**, 12-27 (2024).
38. Leger, A. et al. RNA modifications detection by comparative Nanopore direct RNA sequencing. *Nat Commun* **12**, 7198 (2021).
39. Cerneckis, J., Cui, Q., He, C., Yi, C. & Shi, Y. Decoding pseudouridine: an emerging target for therapeutic development. *Trends Pharmacol Sci* **43**, 522-535 (2022).
40. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094-3100 (2018).

Supplementary Information

Supplementary Figs 1-2.
Supplementary Note 1.
Supplementary Note 2.
Supplementary Note 3.

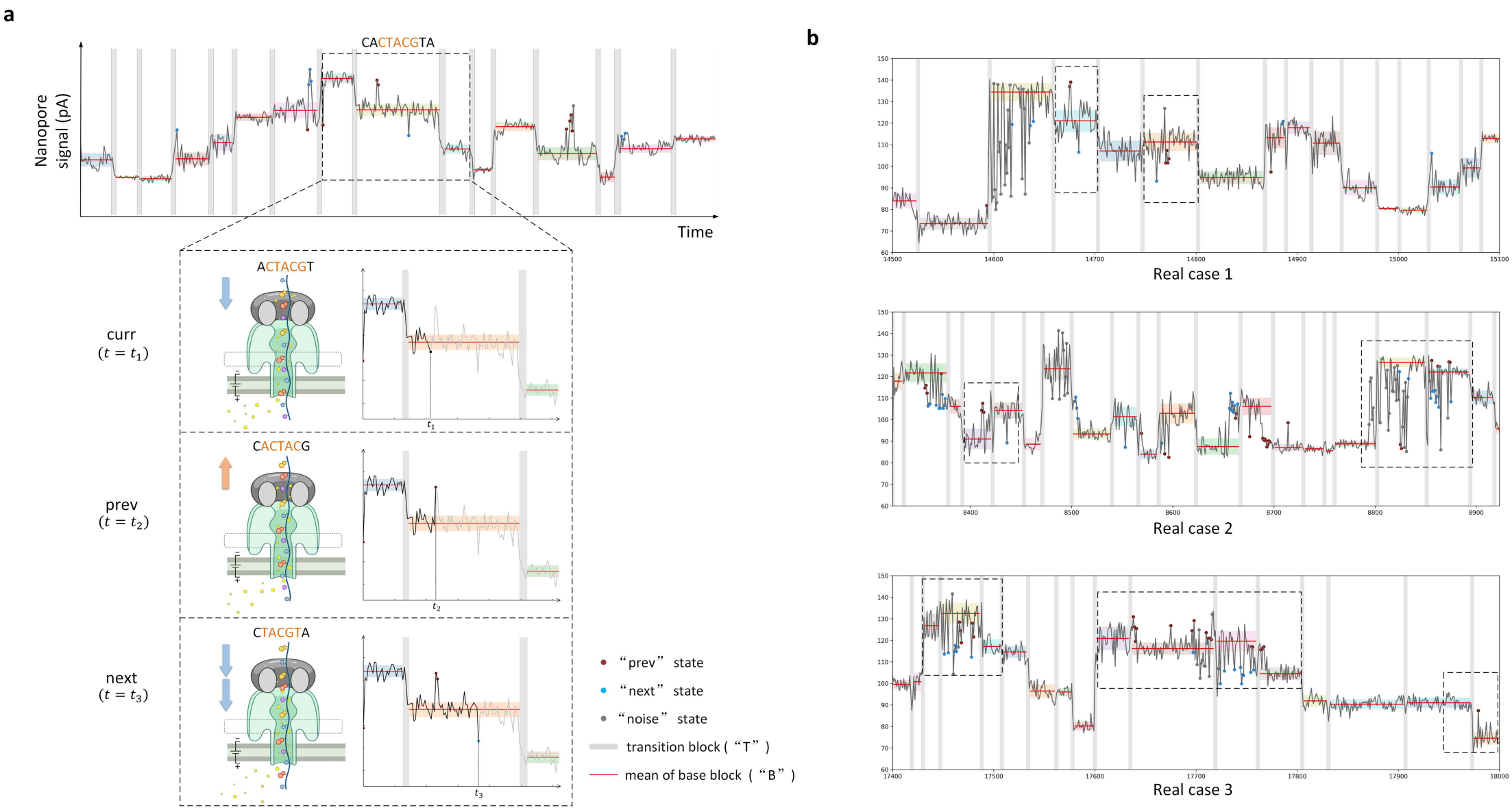

Figure 1: RNA translocation hypothesis. (**a**) Jiggling RNA translocation hypothesis. Top panel shows the raw current signal of Nanopore direct RNA sequencing. The gray areas are SegPore estimated transition blocks. Here we focus on three neighboring 5mers and consider the center 5mer (CTACG) as the current 5mer. The RNA molecule might be translocated forward or backward for a short period during the translocation process of the current 5mer. If the RNA molecule is pulled backward, the previous 5mer ("prev" state) is placed in the pore and the current signal is similar to the previous 5mer's baseline (mean and std highlighted by red lines and shades). If the RNA is pushed forward, the current signal is similar to the next 5mer's baseline ("prev" state). (**b**) Example raw current signals that supports this jiggling hypothesis, with dash rectangles highlighting relevant base blocks. Measurements assigned to the previous and next 5mer are highlighted as red and blue points. It is obvious to observe that red points are close to the previous 5mer's baseline and blue points are close to the next 5mer's baseline. The raw current signals were extracted from mESC WT samples of the training data in the m6A benchmark experiment.

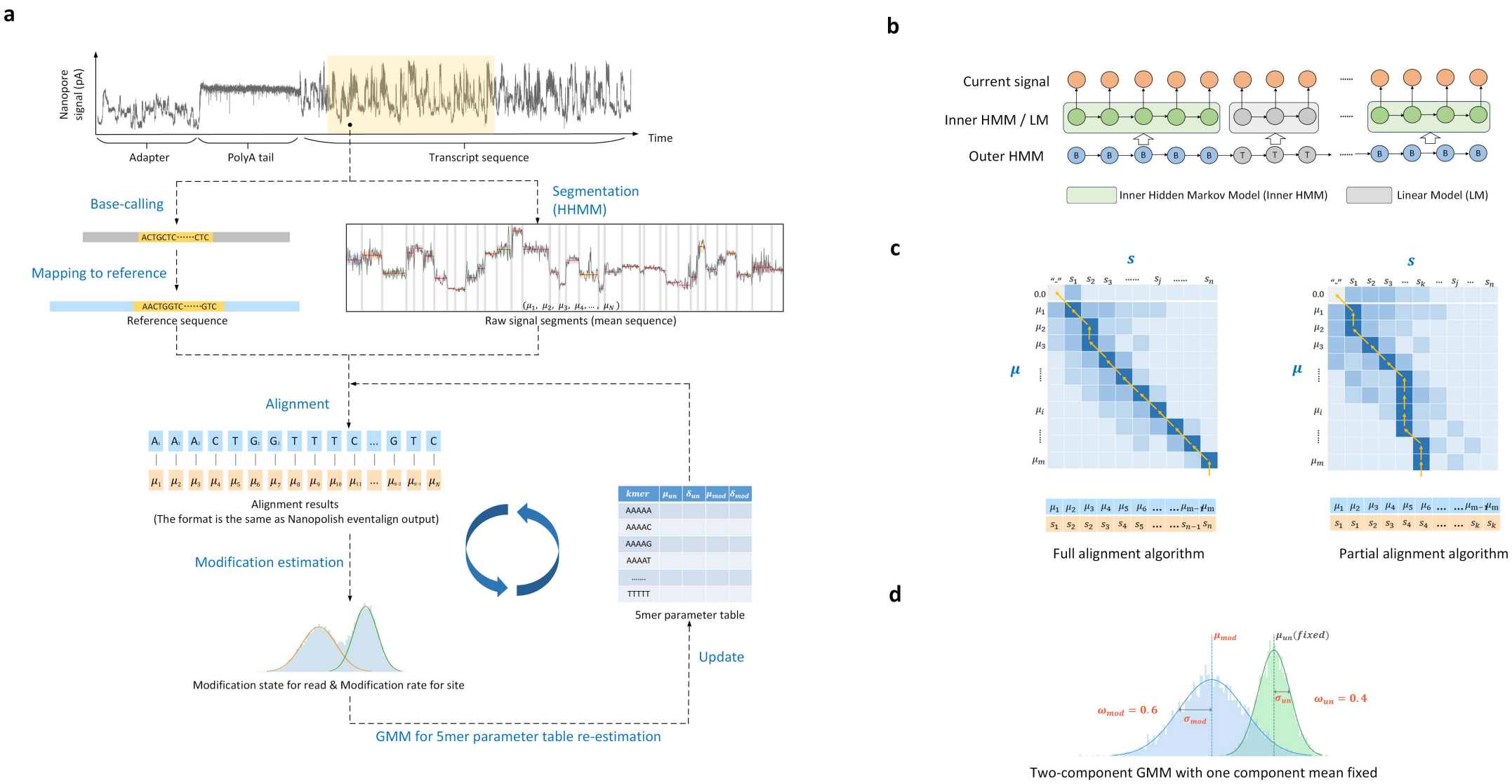

Figure 2: SegPore workflow. (**a**) The general workflow. First, basecalling and mapping are performed using Guppy and Minimap2 such that a raw current signal fragment is paired with a reference sequence fragment. Meanwhile, the raw current signal of a read is split into segments by HHMM and an estimated mean (μ_i) is derived for each segment. Then, the current signal segments (μ_i) are aligned with the 5mer list of the corresponding reference sequence fragment using the full/partial alignment algorithm, given a 5mer parameter table. Here we use A_j to denote A at jth position on the reference. Next, all aligned to the same 5mer at different genomic locations are pooled together and a two-component GMM is fitted to re-estimate the 5mer parameters. One GMM component models the unmodified state and the other models the modified state, while the hidden variable of the GMM specifies the modification state of the 5mer on each read. The parameter estimation process is iterated several times on the training data to gain a stable estimation of the 5mer parameter table. The final 5mer parameter table is used for estimating the modification states on the test data. (**b**) Hierarchical hidden Markov model. The outer HMM partitions current signal into alternating base blocks and transition blocks. An inner HMM approximates the emission probability of a base block by considering neighboring 5mers. A linear model approximates the emission probability of a transition block. (**c**) Full/partial alignment algorithms. Each row is an estimated mean of a base block given by HHMM. Each column is a 5mer of the reference sequence. One 5mer can be aligned to multiple means. (**d**) Gaussian mixture model (GMM) for estimating modification states. The green component codes the unmodified state of a given 5mer. The blue component codes the modified state of the given 5mer. Each component has three parameters: mean (μ), std (δ) and weight (ω).

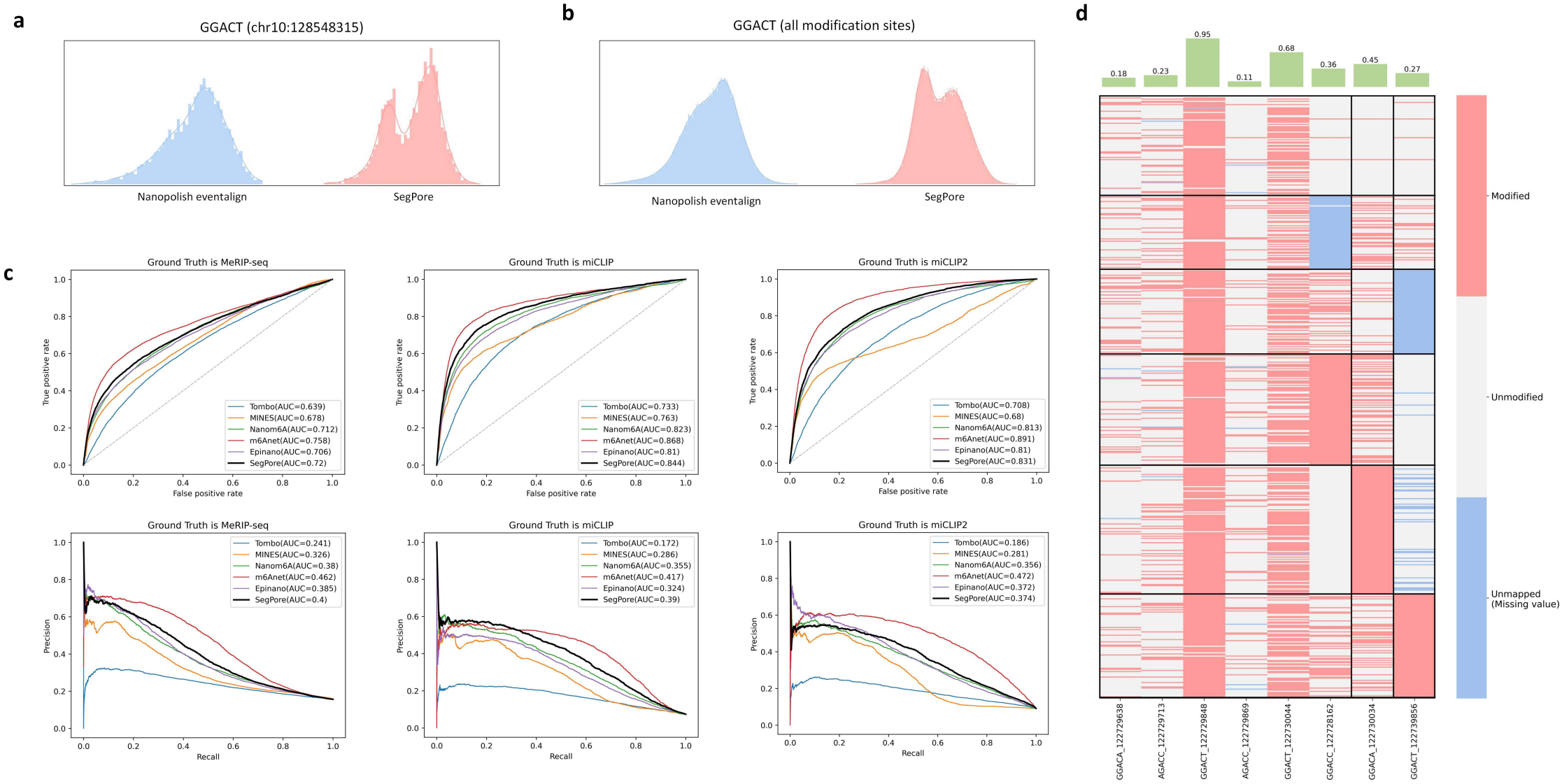

Figure 3: m6A identification. (**a**) Histogram of current signal mapped to an example genomic location (chr10:128548315, GGACT) across all reads in training data given by Nanopolish (left) and SegPore (right). (**b**) Histogram of current signal mapped to GGACT at annotated m6A genomic locations of the training data given by Nanopolish (left) and SegPore (right). (**c**) Benchmark results of Tombo, MINES, Nanom6A, m6Anet, Epinano and SegPore on test data based on different ground truths (MeRIP-seq, miCLIP, miCLIP2). Top panel are receiver operating characteristic (ROC) curves and bottom panel are precision recall (PR) curves. (**d**) Modification rate of selected genomic locations (upper panel) and modification states of all reads mapped to an example gene (ENSMUSG00000003153, lower panel). The black borders in the heatmap highlight the bi-clustering results.

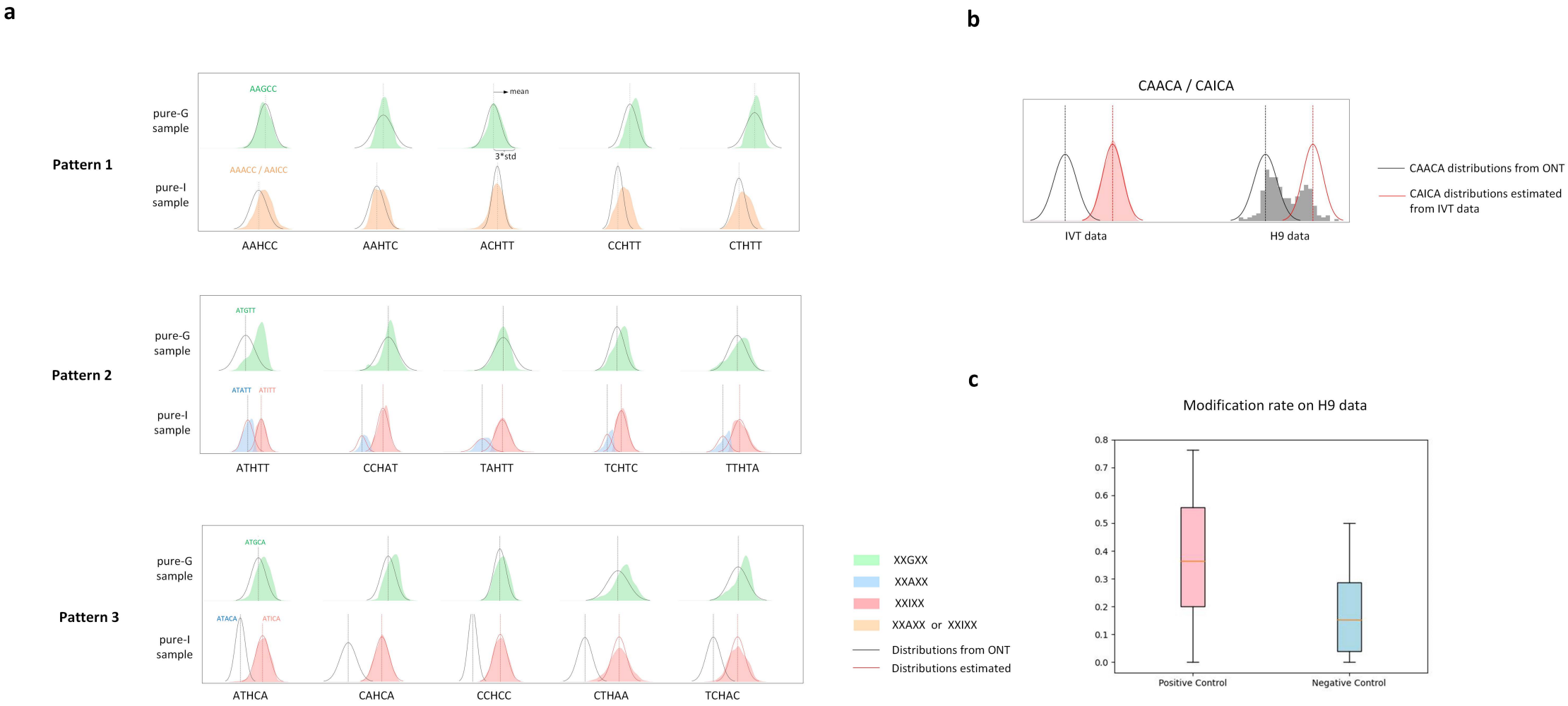

Figure 4: Inosine identification. (**a**) Current signal distributions of example 5mers in IVT data, which are classified into three patterns. The shaded area is the smoothed density of current signal of a given 5mer across different reads and different genomic locations. The fitted Gaussian density curves are given by estimated 5mer parameters. For each pattern, the top panel shows the distributions for 5mers like XXGXX (X is A, C, T/U) in the pure-G sample, while the bottom panel shows distributions of the same 5mers with the center nucleotide replaced by A or I. These examples represent three patterns of 5mers: (1) 5mers that show no difference between A and I (2) 5mers that show significant difference between A and I and (3) 5mers that show significant difference between A and I, but the component weight of A is low. (**b**) Current signal distribution of example 5mers CAACA or CAICA pooled from different reads and different genomic locations in IVT data (training data) and H9 data (test data). (**c**) Modification rate distributions in positive control (inosine modification sites) and negative control (unmodified sites). Modification states are given by the ground truth.

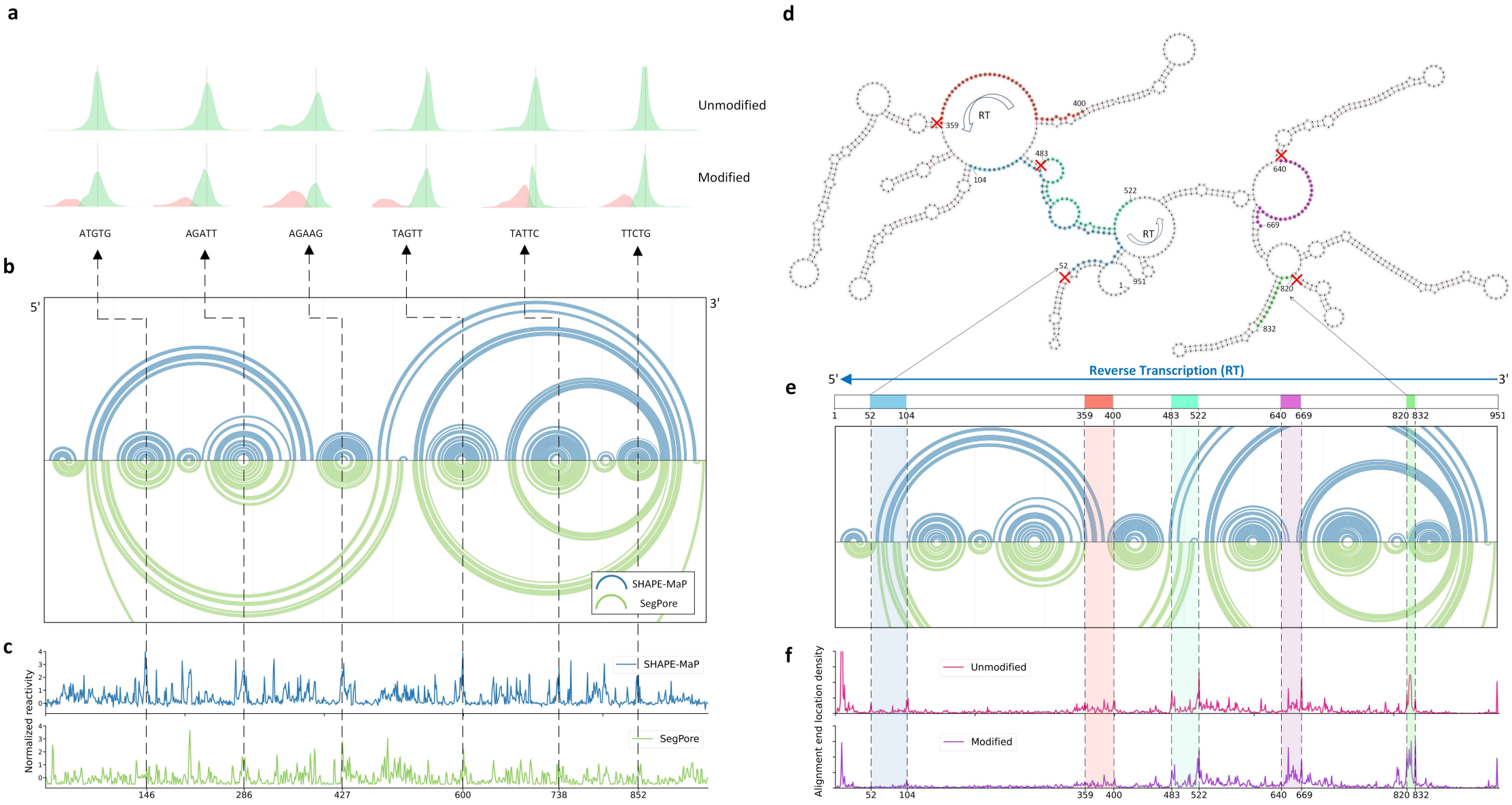

Figure 5: RNA (pri-miR-17~92) structure estimation using SegPore. (**a**) Six example sites chosen from the loop part of six conserved hairpin structures, with the corresponding 5mers displayed at the bottom. The top panel is from the unmodified sample (in vitro transcription) and the lower panel is from the modified sample (adduct treated), whose modified component is highlighted in red. (**b**) Arc diagrams of RNA structure based on SHAPE-Map (blue) and SegPore (green). X-axis is the 5'-3' location on the reference sequence. (**c**) Reactivity scores of SHAPE-MaP and SegPore. A high reactivity score is associated with a high modification rate in the modified sample. (**d**) SegPore predicted RNA structure, with different colors denoting peak regions in the alignment end location distribution. The crossings indicate the stem structures near potential stop locations during reverse transcription. The numbers indicate the 5'-3' locations of the nucleotides. The reverse transcription is from 3' to 5', i.e. from location 951 to 1. (**e**) The same arc diagrams as (b) with peak regions highlighted. (**f**) Alignment end location distribution for both the unmodified and modified samples. X-axis indicates 5'-3' locations on the reference sequence. Y-axis is the probability density.