Detecting m6A RNA modification from nanopore sequencing using a semi-supervised learning framework

Haotian Teng[1 [0000−0003−0337−8722]], Marcus Stoiber[2 [0000−0000−0000−0000]], Ziv Bar-Joseph[1 [0000−0003−3430−6051]], and Carl Kingsford[1 [0000−0002−0118−5516]]

[1] Computational Biology Department, Carnegie Mellon Univeristy, Pittsburgh PA 15213, USA

haotiant@cs.cmu.edu, zivbj@cs.cmu.edu, carlk@cs.cmu.edu

[2] Oxford Nanopore Technologies

Marcus.Stoiber@nanoporetech.com

Abstract. Direct nanopore-based RNA sequencing can be used to detect post-transcriptional base modifications, such as m6A methylation, based on the electric current signals produced by the distinct chemical structures of modified bases. A key challenge is the scarcity of adequate training data with known methylation modifications. We present Xron, a hybrid encoder-decoder framework that delivers a direct methylation-distinguishing basecaller by training on synthetic RNA data and immunoprecipitation-based experimental data in two steps. First, we generate data with more diverse modification combinations through in silico cross-linking. Second, we use this dataset to train an end-to-end neural network basecaller followed by fine-tuning on immunoprecipitation-based experimental data with label-smoothing. The trained neural network basecaller outperforms existing methylation detection methods on both read-level and site-level prediction scores. Xron is a standalone, end-to-end m6A-distinguishing basecaller capable of detecting methylated bases directly from raw sequencing signals, enabling de novo methylome assembly.

Keywords: Nanopore sequencing · m6A RNA modification · Deep learning · hidden Markov model.

RNA modification plays essential roles in various biological processes, including stem cell differentiation and renewal, brain functions, immunity, aging, and cancer progression [1–4]. Among the various types of RNA modifications, N6-Methyladenosine (m6A) is one of the most abundant versions and is involved in various biological processes including mRNA expression, splicing, nuclear exporting, translation efficiency, RNA stability, and miRNA processing [1]. Accurate detection and quantification of m6A modifications is crucial for understanding their impact on gene regulation and cellular processes [5, 6].

Next-generation sequencing (NGS) technologies identify nucleotides through a synthesis process, leading to the loss of post-transcriptional information [7]. Therefore, indirect methods are required to detect RNA modifications with NGS. These approaches first isolate the modified RNA and then conduct reverse transcription and cDNA sequencing to reveal the modifications. Two primary strategies are used to experimentally isolate RNA modifications. One type of approach involves immunoprecipitation. Examples of methods using this approach include MeRIP-Seq [8], m6A-Seq [9], PA-m6A-Seq [10], m6A-CLIP/IP [11], miCLIP [12], m6A-LAIC-Seq [13], m6ACE-Seq [14], and m6A-Seq2 [15]. These methods rely on antibodies that target the modified ribonucleotide and enrich the RNA fragments with the target modified bases. The other type of approach is chemical-based detection. Examples of methods using this approach are Pseudo-Seq [16], AlkAniline-Seq [17], Mazter-Seq [18], m6A-REF-Seq [19], DART-Seq [20], RBS-Seq [21], and m6A-SAC-seq [22]. These techniques use chemical compounds or enzymes that selectively interact with the modified ribonucleotide, either cleaving or modifying the RNA reads to halt or disturb the reverse transcription process. This is followed by short-read cDNA sequencing, which identifies the RNA modifications by comparing the read ends of the cDNA or the base mismatches/deletions in cDNA. Although these methods were able to generate detailed maps of RNA modification sites, they all use external compounds which makes it hard to obtain the required single base resolution. They also face other challenges and shortcomings including the limited availability of antibodies or compounds for specific modifications [23], nonspecific antibody binding [24–26], low single-nucleotide resolutions [8, 9], and, importantly, an inability to identify the exact location of a modification.

Direct RNA sequencing using nanopores offers a promising alternative [27]. An RNA molecule can be sequenced by measuring the intensity of the current flowing through the pore as the RNA molecules pass through it. Modified RNA nucleotides produce different signals than their unmodified counterparts, providing information about the modifications at the single-molecule read resolution [28, 29]. However, to detect specific modifications from subtle signal changes we need an optimized algorithm, which is normally obtained through supervised learning or a comparative approach [30]. Unfortunately, current data are not immediately suitable for supervised learning due to the lack of experimental techniques for identifying the methylation state at the single-read resolution.

In vitro transcription (IVT) data, which are transcribed from either experimentally synthesized DNA sequences or native DNA [28, 31], can provide reads that are either completely methylated or not methylated at all (all-or-none), but the diversity of the sequence compositions in synthesized DNA datasets is limited due

to the constraints concerning the maximum DNA length that can be synthesized and the associated costs. In addition, the IVT dataset lacks partially methylated reads with known methylation states. Although partially methylated reads can be generated by introducing a mixture of modified and canonical adenine during *in vitro* transcription, the location of methylation remains unknown because in such mixtures the RNA polymerase randomly selects adenine from either type during the transcription process. Models trained to identify modifications on all-or-none modified reads perform poorly on biological reads, which are usually sparsely methylated [31, 32]. Methods using such synthesized datasets include training a classifier to predict sequence segments (5-mers) given their corresponding nanopore raw signal segments [33] or features of these segments [28, 29, 31, 34]. The signal segments are extracted from raw signal after performing base-calling and alignment, using models trained on canonical data (data with no methylation). As we show, the performance of such a classifier is limited since it is only trained on isolated short segments, losing contextual information. In addition, these models are trained solely on manually selected features including mean, standard deviation, and duration of isolated signal segments corresponding to 5 bases, which can lead to the loss of more detailed signal information. Recently, a new method, CHEUI, was trained using longer signal segments, yielding impressive results on IVT data [35]. However, it suffers from overfitting when applied to real biological samples (Fig. 2, [36]).

Immunoprecipitation (IP) data from assays such as m6ACE-seq and m6A-CLIP-seq relies on the use of antibodies [11, 12, 37]. However, this strategy works at a high level. It only provides the modification proportion for each reference transcriptomic position, i.e., a site-level modification rather than the modification state for each individual read (read-level). m6Anet [36] employs multiple-instance learning [38] to train a classifier using IP data leading to improved site-level accuracy. However, IP data misses many methylation sites, particularly in low-coverage regions [25]. Additionally, due to nonspecific antibody binding, the methylation detection results obtained through immunoprecipitation experiments produced a false-positive rate of approximately 11%, which can vary between studies [18, 39]. M6Anet also requires a minimum coverage level of 20 reads for a site to be detected due to the way the model is trained. The training involves maximizing the probability of detecting at least one methylated read among the reads covering a known methylated site. Such coverage depth is not always available. Finally, as in the other existing models, m6Anet relies on a basecaller and segmentation tools that are trained on nonmodified reads (canonical reads).

In summary, previous approaches try to identify m6A sites using basecalling errors [28, 29, 31, 34], by comparing between control samples [29, 40], trained on IVT data [33, 35] or trained on noisy labels from IP

data [36]; these methods are summarized in Tables 1 and S1. As we will show, the fact that they are only trained on one type of data limits their performance (Figure. 2a,b and Supplementary Figure. 3).

We present a method that takes a different approach by detecting methylation during the basecalling phase. We predict methylated bases directly from the long current signal by training a methylation-distinguishing basecaller. To achieve this, we developed Xron, a hybrid encoder-decoder framework (Fig. 1). The encoder is a convolutional recurrent neural network (CRNN) encoding the observable signal into a kmer representation. After it has been trained and fine-tuned, the CRNN serves as a methylation-distinguishing basecaller for new data. The decoder is a nonhomogeneous hidden Markov model (NHMM), which serves as a generative model for achieving signal segmentation and alignment when preparing the training dataset. Applying the NHMM, we created a partially methylated dataset to train the CRNN and produce a methylation-distinguishing basecaller. The CRNN is then fine-tuned using the IP data, further enhancing the basecaller's generalizability. This framework enables us to obtain a highly accurate methylation-distinguishing basecaller by exploiting both IVT data and IP data. This approach outperforms all previous methods on synthesized and biological samples and provides a comprehensive, end-to-end solution for methylation base detection.

Table 1. Reported Performance of m6A Modification Identification Achieved by Existing Works

	AUC ROC			
Method	*Read-level	*Site-level	Yeast KO[31]	Human[41]
Epinano (2019) [31]	–	0.90	0.680	–
ELIGOS (2021) [28]	–	0.756	0.287 (F1)	–
Nanocompore (2021) [29]	–	–	0.18 (F1)	–
nanom6A (2021) [33]	–	0.97	0.71	–
CHEUI (2022) [35]	0.806	0.92	–	–
m6Anet (2022) [36]	0.90	0.94	–	0.83
Xron (this work)	**0.93**	**>0.99**	**0.90**	**0.91**

*These results were reported on the IVT dataset [31], in which single-read m6A modifications were known.

Results

Applying Xron to identify m6A methylation on direct RNA sequencing datasets

Xron performs methylation-distinguishing basecalling, outputting methylated bases directly from the raw sequencing signal emitted from the nanopore. Its neural network basecaller is trained on an augmented partially methylated dataset and then fine-tuned using IP data. We tested Xron on three public direct

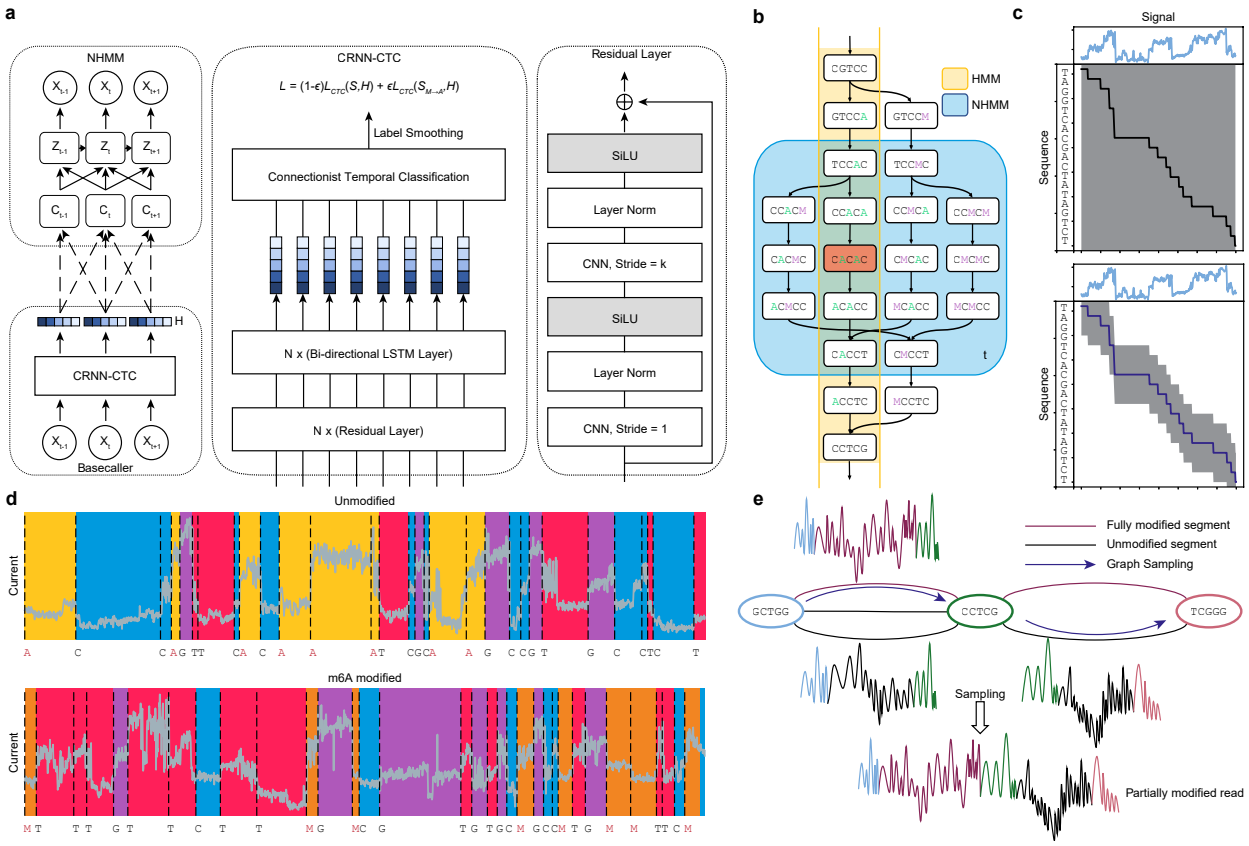


Fig. 1. Schematics of Xron model and the data augmentation process through crosslinking and sampling. a, Xron consists of two parts: a nonhomogeneous hidden Markov model (NHMM) and a convolutional recurrent neural network (CRNN) with a connectionist temporal classification (CTC) decoder. **b**, Comparison between HMM and NHMM. The transition matrix of a HMM (yellow) encodes the whole Markov chain of k-mers, while the transition matrix of the NHMM (blue) at time t only encodes the Markov chain of the five nearby k-mers given the predicted k-mer (shown in red) at time t. The Markov chain is also expanded to include the k-mers with all combinations of the A and M (m6A) bases. We create partially methylated reads using data augmentation, first segmenting the signal and then cross-linking the reads and their corresponding signal in silico. To achieve this, we design a novel nonhomogeneous hidden Markov model (NHMM) that can be trained to conduct signal segmentation in a semi-supervised fashion on modified reads, even when lacking methylation labels. The NHMM is trained using the forward-backward algorithm with its transition matrix conditioned on a canonical baseaclled sequence and its alignment, thus giving the maximum likelihood estimation of the model parameters regarding methylation base. The Viterbi path of the NHMM gives the alignment between the current signal and sequence. Following the signal segmentation process performed with the NHMM, the NHMM was used to create a training dataset with partially methylated reads and their true labels for methylation detection training by augmenting all-or-none modified reads. **c**, The transition process of the NHMM is constrained by the neural network's output, leading to a smaller probability space and making it easier for the model to find the optimal alignment. **d**, The NHMM is trained in a semi-supervised manner on IVT datasets, including fully modified, unmodified, and partially modified reads. It provides accurate signal segmentation results for both unmodified and modified sequences. **e**, In-silico read crosslinking. The fully modified or unmodified reads are first broken into segments at the invariant k-mers to form a signal-k-mer graph, whose nodes are k-mers and whose edges are signal segments. Then, a partially methylated read is sampled from the k-mer signal graph.

RNA sequencing datasets: an IVT dataset [31], a yeast dataset [31], and a human embryonic kidney cells (HEK293T) dataset [36].

The IVT dataset [31] was synthesized from artificially designed sequences followed by *in vitro* transcription. The dataset contains either fully methylated or fully unmethylated reads. Signal intensity shows differences around the center base of the kmer between modified and unmodified sites (Fig. 3a and Supplementary Fig. 1). The sequences are designed to contain all 5-mers, including the most common k-mer (GGACT) and all 18 DRACH motifs (Fig. 3a,b).

The yeast dataset [31] contains direct RNA sequencing reads from two strains, a wild-type strain, and a "$ime4\Delta$" knockout strain, in which IME4 was deleted. The deletion of IME4 results in the complete elimination of m6A bases, making it a negative control. The yeast dataset contains three independent biological replicates for each strain. Two were used in this study; the first replicate was used for training, and the second was used for evaluation.

The human HEK293T cell dataset [36] contains direct RNA-Seq data from the HEK293T cell line [34], with methylation sites identified by m6ACE-Seq [14] and miCLIP data [12] on the same cell line. The dataset contains three replicates, and we used the first replicate to evaluate the method. (See Methods for details about replicates and datasets used for training and evaluation.)

Xron accurately identifies m6A sites

To evaluate the performance of Xron, we applied Xron that is finetuned on yeast data to direct RNA sequencing data derived from the human HEK293T cell line [34]. Although Xron is pre-trained using human IVT reads (Methods), no human methylation information is used during training since all human reads are canonical. To validate the model, we used the m6A sites detected by m6ACE-Seq and miCLIP from the human HEK293T cell line as the true labels during evaluation, following previous work [36]. We used the m6A sites identified by m6ACE-Seq and miCLIP as positive samples and the other sites with the same 5-mer as negative samples. Xron achieved the best ROC AUC of 0.91 (Fig. 2a) compared with those of Epinano (0.69) and m6Anet (0.83) and the best precision-recall (PR) AUC of 0.456 (Fig. 2a) compared to m6Anet (0.342) and MINES (0.256).

Xron is sensitive to IME4 knockouts

In addition, we also evaluated Xron on a yeast dataset using a *ime4Δ* knockout *S. cerevisiae* strain where the m6A modification was completely eliminated [37] as the control dataset, following a previous study [31]. We used the second replicate sample of the dataset for evaluation, as we had fine-tuned Xron on a subset of the first replicate. We treated the m6A sites in the wild-type strain as modified sites and the same sites in the *ime4Δ* knockout strain as unmodified sites. We compared Xron with other models for predicting modified/unmodified sites. Xron achieved an AUC-ROC score of 0.90 (Fig. 2b) on this task, providing a 21% increase over the second-best model, Epinano (0.72).

Xron detects more methylation sites and achieves high accuracy under low coverage settings

As m6anet intrinsically requires a minimum coverage of at least 20 to obtain site methylation predictions, this results in a much smaller sample size (11 sites detected). In the same setting, Xron yields 171 sites with a minimum coverage of 20 on the yeast dataset, which results in higher AUC-ROC accuracy than m6anet (0.90 versus 0.69). In total, Xron detects 272 sites reported in the IP data, compared to the 156 sites detected by Epinano and the 93 sites detected by CHEUI (Fig. 2c). Sites detected by Xron also show higher support from the IP technique (124) compared to m6Anet (107) in the HEK293T cell line (Fig. 2d). While different methods identify various m6A methylation sites, many sites are detected exclusively by one method. This observation aligns with previous reports [14, 36]. We next tested if including more low-coverage sites by setting different minimum sequencing coverage thresholds would influence the prediction accuracy of Xron (Fig. 2e). We found that increasing the read coverage yielded superior site-level methylation prediction accuracy, increasing from a 0.825 AUC-ROC score for a minimum read coverage level of 4 to a 0.930 AUC-ROC score with a minimum read coverage level of 28. This suggests that with higher sequencing depth, Xron can further enhance the precision and accuracy of methylation detection. Meanwhile, Xron outperforms other models by a large margin even when setting the minimum read coverage level to 4, with AUC 14% more than the second best model, Epinano (0.825 versus 0.72). Furthermore, to evaluate Xron's performance in low-coverage regions, we down-sampled the reads to limit the maximum coverage at each site to a range of 10 to 70. Xron achieved an accuracy of 0.725 with maximum coverage of 10, outperforming other models with full data (Fig. 2f,g).

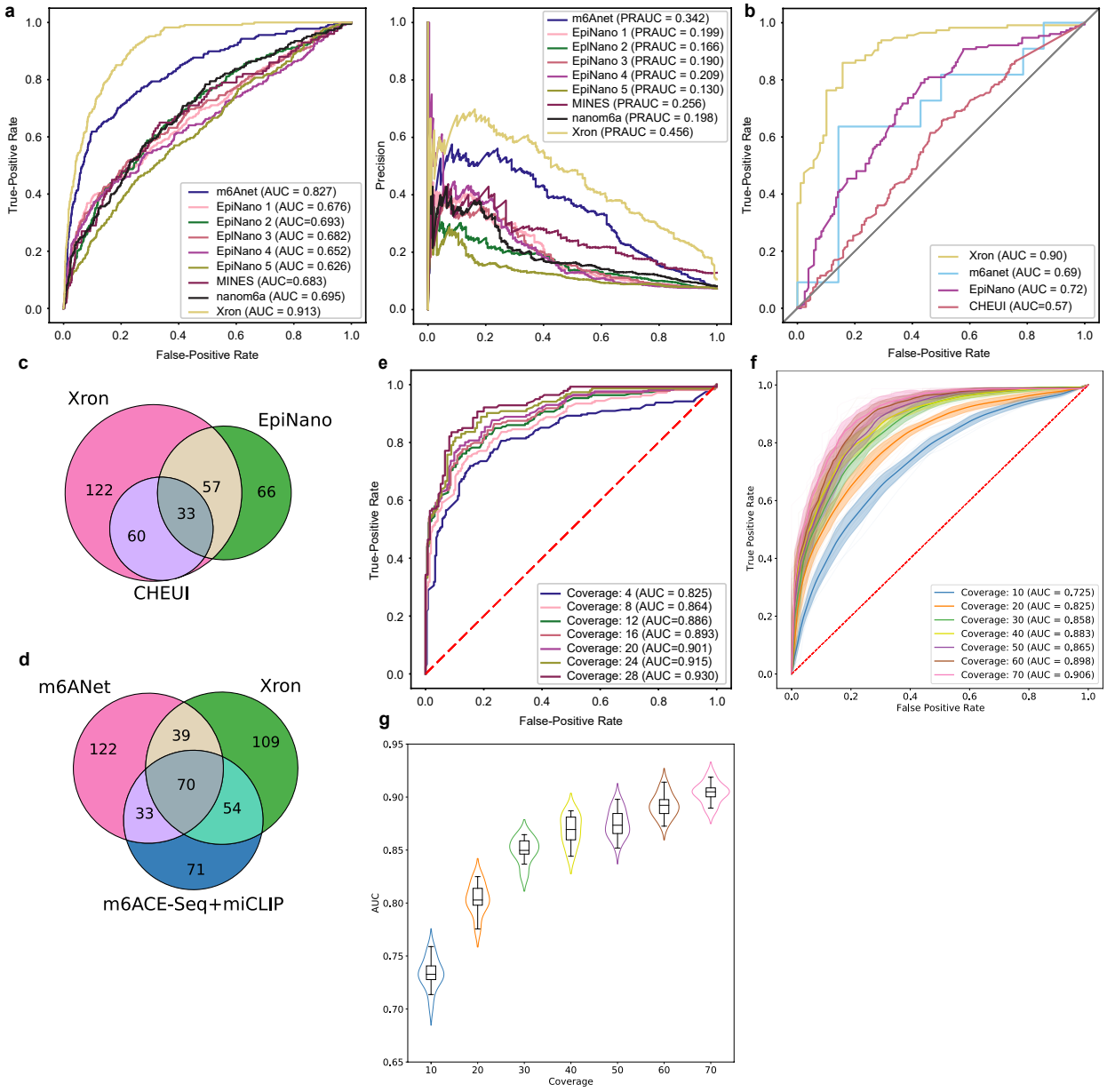


Fig. 2. Comparison of Xron models across two different species. a, ROC and PR curves of m6A prediction on human HEK293T cell line, produced by Xron and other models. **b**, ROC curves produced by Xron and other models on yeast data. **c**,**d**, Venn diagram showing the overlapping sites predicted by Xron and other methods on Yeast (**c**) and HEK293T (**d**) data. **e**, ROC curves produced by Xron for detecting m6A methylation in yeast data under different minimum sequence coverage thresholds. **f**, ROC curves generated by Xron for detecting m6A methylation in down-sampled yeast data with different coverage. **g**, Distribution of AUC score of Xron on down-sampled yeast data.

Xron achieves nearly optimal site-level prediction on a synthesized RNA dataset

We evaluated Xron on a synthesized RNA IVT dataset [31] obtained from a different replicate than the training dataset (see the Methods section). In this dataset, the true methylation modifications were known

for each position in each read, as the reads were either from a fully modified or a fully unmodified run. Our model achieved an AUC ROC of 0.93 on the single-read-level prediction task (Fig. 3c), in which the model has to predict m6A bases or A bases for each read at RRACH sites identified by previous antibody immuno-precipitation experiments [37]. Our model outperforms the second-best read-level model (m6anet) by 3% (0.93 versus 0.90) and an almost optimal AUC ROC of >0.99 for site-level prediction (Fig. 3d), outperforming the second-best site-level model (CHEUI) by nearly 2% (≈ 1 versus 0.98).

Xron provides m6A stoichiometry

By aligning the reads to the reference genome and piling up the single-read m6A modification predictions for different sites, Xron can predict site-level m6A modification stoichiometry, i.e., the fraction of modified bases at a site. We evaluated this ability using a synthetic dataset.

The dataset was a mixture created by randomly sampling reads from fully modified or unmodified IVT datasets [31] with specific mixture proportions, which included 0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100%. We calculated the model-predicted m6A proportion as the number of m6A bases called per site divided by the total number of reads aligned to this site. The median relative modification proportion followed the same trend as the expected methylation proportion. The trend in stoichiometry level was successfully recovered (Fig. 3e).

Xron performs consistent basecalling on m6A-modified datasets

To compare the performance of Xron as a basecaller with a canonical basecaller, we evaluated the basecalling accuracy of Xron and compared it with that of the Guppy ONT basecaller (Table 2 and Supplementary Table S2). We evaluated the basecall quality achieved on three datasets: the synthesized IVT RNA dataset, the *S. cerevisiae* yeast dataset, and the human HEK293T cell line dataset, considering both modified and unmodified reads. For the synthesized IVT RNA and yeast datasets, we used the second replicate, which was not used as training data. Xron suffers less (or no) accuracy drop on datasets with m6A modifications. It exhibited no performance loss on datasets with methylation compared to the control dataset. On the other hand, Guppy showed performance decreases on all three datasets with methylation compared to its performance on the unmodified control datasets, including a 14.47% drop in the identity rate on the synthesized reads and a 7.55% drop in the identity rate on the HEK293T reads.

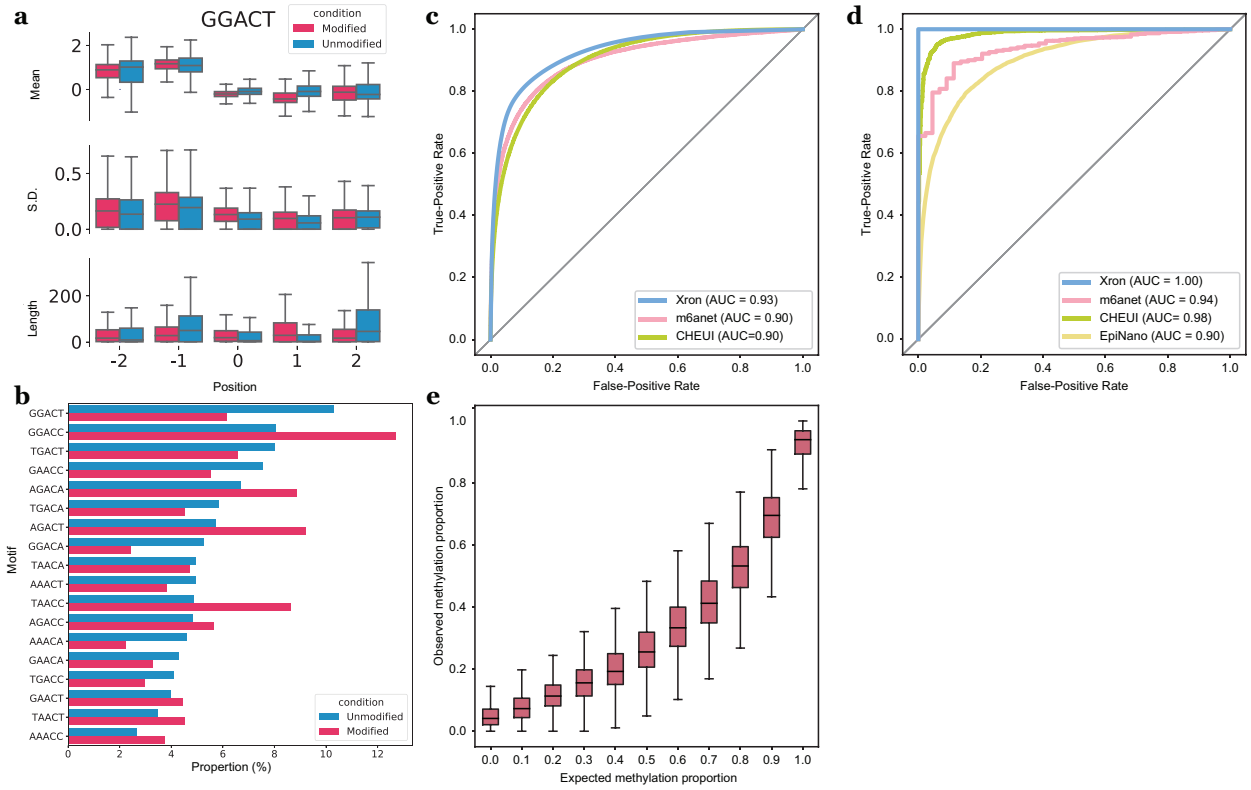


Fig. 3. Evaluation of the m6A detection results obtained for synthesized IVT RNA reads and stoichiometry prediction. a, Box plot comparing the distribution of the mean, standard deviation, and length for the signal segmented by NHMM with 5,232 modified sites and 18,464 unmodified sites for the GGACT motif. Horizontal lines show the median, the box denotes the interquartile range, and the whiskers extend to 1.5 times the interquartile range. Points beyond this range are considered outliers and are removed from the plot. **b**,**c**, ROC curves of Xron against m6anet and CHEUI for read-level (**b**) and site-level (**c**) m6A modification predictions. **d**, Bar plot showing the relative proportion of DRACH 5-mer motif for 84,919 modified and 179,717 unmodified positions. **e**, Box plot showing the m6A ratio predicted by Xron with different proportions of IVT control and IVT m6A RNA mixing.

Discussion

Several computational methods [28, 29, 31, 33, 35] have been used to detect m6A methylation. These methods require accurate training data, usually obtained using synthesized RNA reads containing the modification of interest, obtained through experimental methods such as m6ACE-Seq or miCLIP, or from a comparative analysis against control data. However, these methods exhibit a performance drop when they are applied to other datasets, implying the existence of overfitting. In addition, these methods usually can only provide site-level methylation, losing read-level resolution. We developed an end-to-end m6A modification detection

Table 2. Accuracy comparison between Xron and Guppy on three different datasets and their control datasets. The identity rate (%) was defined as the number of matched bases in the query sequence divided by the number of bases in the reference sequence (the higher the better). All reported rates are mean values among the aligned reads.

Condition	**Model**	Identity rate (%)(↑)	Identity rate change (%)
IVT Control	Xron	87.35	—
	Guppy	92.75	—
IVT m6A	Xron	88.48	1.13
	Guppy	78.28	−14.47
Yeast ime4Δ KO	Xron	83.42	—
	Guppy	92.50	—
Yeast	Xron	83.96	0.54
	Guppy	91.94	−0.56
HEK293T Mettl3 KO	Xron	85.91	—
	Guppy	93.19	—
HEK293T	Xron	87.12	1.21
	Guppy	85.64	−7.55

system for nanopore direct RNA sequencing and, for the first time, created an m6A-distinguishing base caller. Our system, Xron, includes an NHMM model for k-mer decoding and a neural network basecaller. By employing data augmentation and semi-supervised learning, we constructed an NHMM that is capable of performing accurate signal sequence alignment and introduced a novel training dataset for m6A methylation detection. The training pipeline established in our work facilitates supervised basecaller training without necessitating complex feature engineering and using both IVT and IP data available to overcome overfitting.

Quantifying the transcriptome-wide modification rates is one of the key challenges in methylation detection. From the read-level methylation states given by Xron, the modification stoichiometry for each site can be obtained. Meanwhile, our method does not require a high minimum coverage depth, which is essential for detecting methylation in low-expression regions. Comparative methods detect methylation by analyzing data from different conditions [29, 34]. While Xron does not require a control sample to detect methylation, it can also facilitate the use of a control sample by comparing the same site across samples. In addition, compared to other methods where the model performance is influenced by aspects such as base-calling algorithms, accuracy in the alignment of the reference sequence to signal, and segmentation of the raw signal, Xron reads out methylation information directly from the raw signal. More training data on different experimental protocols and different organisms will likely further improve the accuracy of Xron and other supervised approaches, while the training framework of Xron can easily adopt these additional training data into the finetuning pipeline.

As a basecaller, Xron achieves a consistent identity rate among methylation and unmethylation datasets. Although there is a performance gap in terms of identity rate between Xron and the basecaller Guppy, this is likely due to the different neural network architecture used. In future research, it would be beneficial to investigate various neural network structures since previous studies have shown that alterations to the convolutional-recurrent neural network architecture can yield enhanced basecalling accuracy. For example, Guppy uses QuartzNet [42], a neural network designed initially for speech recognition. SACall [43] employs an attention mechanism, while RODAN [44] integrated squeeze-and-excitation [45] layers into a base CNN.

Currently, the NHMM takes only raw signal as its input. This has several advantages, including being easy to train and having a closed-form solution for parameter estimation. However, additional input features can be added to the NHMM, including the encoded representation from the neural network base caller.

Xron was used to detect m6A modification, however, our framework is suitable for training a basecaller for detecting any natural post-transcription modification, including DNA methylation such as m5C and other types of RNA modification. Xron can also be retrained to detect artificial modifications at a single-molecule level, such as detecting modifications introduced in small non-coding RNA [46].

Acknowledgements

This work was supported in part by the US National Science Foundation [DBI-1937540, III-2232121], the US National Institutes of Health [R01HG012470], and by the generosity of Eric and Wendy Schmidt by recommendation of the Schmidt Futures program. We also thank the Pittsburgh Supercomputing Center for providing computational resources through the Bridges2 system. H.T. is supported by funding from Oxford Nanopore Technologies plc and the School of Computer Science, Carnegie Mellon University - The Joint CMU-Pitt Ph.D. Program in Computational Biology (CPCB). Conflict of Interest: C.K. is a co-founder of Ocean Genomics, Inc. H.T. is supported by funding from Oxford Nanopore Technologies plc. M.S. is an employer of Oxford Nanopore Technologies plc. We thank Minh Hoang for reviewing the manuscript and offering valuable feedback. We thank Tim Massingham (XGenomes Corp.) for the helpful discussion on signal segmentation.

Code Availability

Code is hosted at GitHub repository https://github.com/haotianteng/xron. Xron is available under a GNU GENERAL PUBLIC LICENSE v3.0. Xron is built with Python 3.8 and PyTorch 1.12, and has been tested

on PyTorch 1.13 and 2.0. We used ChatGPT to correct grammatical errors and improve the flow of early drafts of this manuscript.

Data Availability

The IVT RNA datasets were obtained from Epinano project [31] through the GEO database (GSE124309). The ELIGOS IVT RNA datasets were obtained from ELIGOS project [28] through the SRA database (SRP166020). The Yeast datasets (wild and ime4-knockout) were obtained from Epinano Project [31] through the GEO database (GSE126213). The HEK293T cell lines data were obtained from the SG-NEx Project [41] through ENA (PRJEB40872).

References

1. Boulias, K. & Greer, E. L. Biological roles of adenine methylation in RNA. *Nature Reviews Genetics* **24,** 143–160 (2023).
2. Sun, T., Wu, R. & Ming, L. The role of m6A RNA methylation in cancer. *Biomedicine & Pharmacotherapy* **112,** 108613 (2019).
3. D'Aquila, P. *et al.* Methylation of the ribosomal RNA gene promoter is associated with aging and age-related decline. *Aging Cell* **16,** 966–975 (2017).
4. Qin, Y. *et al.* Role of m6A RNA methylation in cardiovascular disease. *International Journal of Molecular Medicine* **46,** 1958–1972 (2020).
5. Murakami, S. & Jaffrey, S. R. Hidden codes in mRNA: Control of gene expression by m6A. *Molecular Cell* **82,** 2236–2251 (2022).
6. Fu, Y., Dominissini, D., Rechavi, G. & He, C. Gene expression regulation mediated through reversible m6A RNA methylation. *Nature Reviews Genetics* **15,** 293–306 (2014).
7. Buermans, H. & Den Dunnen, J. Next generation sequencing technology: advances and applications. *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease* **1842,** 1932–1941 (2014).
8. Meyer, K. D. *et al.* Comprehensive analysis of mRNA methylation reveals enrichment in 3' UTRs and near stop codons. *Cell* **149,** 1635–1646 (2012).
9. Dominissini, D. *et al.* Topology of the human and mouse m6A RNA methylomes revealed by m6A-seq. *Nature* **485,** 201–206 (2012).
10. Chen, K. *et al.* High-resolution N6-methyladenosine (m6A) map using photo-crosslinking-assisted m6A sequencing. *Angewandte Chemie* **127,** 1607–1610 (2015).

11. Ke, S. *et al.* A majority of m6A residues are in the last exons, allowing the potential for 3' UTR regulation. *Genes & Development* **29,** 2037–2053 (2015).

12. Linder, B. *et al.* Single-nucleotide-resolution mapping of m6A and m6Am throughout the transcriptome. *Nature Methods* **12,** 767–772 (2015).

13. Molinie, B. *et al.* m6A-LAIC-seq reveals the census and complexity of the m6A epitranscriptome. *Nature Methods* **13,** 692–698 (2016).

14. Koh, C. W., Goh, Y. T. & Goh, W. Atlas of quantitative single-base-resolution N6-methyl-adenine methylomes. *Nature Communications* **10,** 1–15 (2019).

15. Dierks, D. *et al.* Multiplexed profiling facilitates robust m6A quantification at site, gene and sample resolution. *Nature Methods* **18,** 1060–1067 (2021).

16. Carlile, T. M. *et al.* Pseudouridine profiling reveals regulated mRNA pseudouridylation in yeast and human cells. *Nature* **515,** 143–146 (2014).

17. Marchand, V. *et al.* AlkAniline-Seq: profiling of m7G and m3C RNA modifications at single nucleotide resolution. *Angewandte Chemie International Edition* **57,** 16785–16790 (2018).

18. Garcia-Campos, M. A. *et al.* Deciphering the "m6A code" via antibody-independent quantitative profiling. *Cell* **178,** 731–747 (2019).

19. Zhang, Z. *et al.* Single-base mapping of m6A by an antibody-independent method. *Science Advances* **5,** eaax0250 (2019).

20. Meyer, K. D. DART-seq: an antibody-free method for global m6A detection. *Nature Methods* **16,** 1275–1280 (2019).

21. Khoddami, V. *et al.* Transcriptome-wide profiling of multiple RNA modifications simultaneously at single-base resolution. *Proceedings of the National Academy of Sciences of the United States of America* **116,** 6784–6789 (2019).

22. Hu, L. *et al.* m6A RNA modifications are measured at single-base resolution across the mammalian transcriptome. *Nature Biotechnology* **40,** 1210–1219 (2022).

23. Ryvkin, P. *et al.* HAMR: high-throughput annotation of modified ribonucleotides. *RNA* **19,** 1684–1692 (2013).

24. Helm, M., Lyko, F. & Motorin, Y. Limited antibody specificity compromises epitranscriptomic analyses. *Nature Communications* **10,** 5669 (2019).

25. McIntyre, A. B. *et al.* Limits in the detection of m6A changes using MeRIP/m6A-seq. *Scientific Reports* **10,** 6590 (2020).

26. Zhang, Z. *et al.* Systematic calibration of epitranscriptomic maps using a synthetic modification-free RNA library. *Nature Methods* **18,** 1213–1222 (2021).

27. Garalde, D. R. *et al.* Highly parallel direct RNA sequencing on an array of nanopores. *Nature Methods* **15,** 201–206 (2018).

28. Jenjaroenpun, P. *et al.* Decoding the epitranscriptional landscape from native RNA sequences. *Nucleic Acids Research* **49,** e7–e7 (2021).

29. Leger, A. *et al.* RNA modifications detection by comparative Nanopore direct RNA sequencing. *Nature Communications* **12,** 1–17 (2021).

30. Wan, Y. K., Hendra, C., Pratanwanich, P. N. & Göke, J. Beyond sequencing: machine learning algorithms extract biology hidden in Nanopore signal data. *Trends in Genetics* **38,** 246–257 (2022).

31. Liu, H. *et al.* Accurate detection of m6A RNA modifications in native RNA sequences. *Nature Communications* **10,** 1–9 (2019).

32. Zhong, Z.-D. *et al.* Systematic comparison of tools used for m6A mapping from nanopore direct RNA sequencing. *Nature Communications* **14,** 1906 (2023).

33. 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 Biology* **22,** 1–17 (2021).

34. Pratanwanich, P. N. *et al.* Identification of differential RNA modifications from nanopore direct RNA sequencing with xPore. *Nature Biotechnology* **39,** 1394–1402 (2021).

35. Mateos, P. A. *et al.* Identification of m6A and m5C RNA modifications at single-molecule resolution from Nanopore sequencing. *bioRxiv https://doi.org/10.1101/2022.03.14.484124* **14** (2022).

36. Hendra, C. *et al.* Detection of m6A from direct RNA sequencing using a multiple instance learning framework. *Nature Methods* **19,** 1590–1598 (2022).

37. Schwartz, S. *et al.* High-resolution mapping reveals a conserved, widespread, dynamic mRNA methylation program in yeast meiosis. *Cell* **155,** 1409–1421 (2013).

38. Amores, J. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence* **201,** 81–105 (2013).

39. Ke, S. *et al.* m6A mRNA modifications are deposited in nascent pre-mRNA and are not required for splicing but do specify cytoplasmic turnover. *Genes & Development* **31,** 990–1006 (2017).

40. Abebe, J. S. *et al.* DRUMMER—rapid detection of RNA modifications through comparative nanopore sequencing. *Bioinformatics* **38,** 3113–3115 (2022).

41. Chen, Y. *et al.* A systematic benchmark of Nanopore long read RNA sequencing for transcript level analysis in human cell lines. *BioRxiv,* 2021–04 (2021).

42. Kriman, S. *et al. Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions* in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), 6124–6128.

43. 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 Transactions on Computational Biology and Bioinformatics* **19,** 614–623 (2020).

44. Neumann, D., Reddy, A. S. & Ben-Hur, A. RODAN: a fully convolutional architecture for basecalling nanopore RNA sequencing data. *BMC Bioinformatics* **23,** 1–9 (2022).

45. Hu, J., Shen, L. & Sun, G. *Squeeze-and-excitation networks* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), 7132–7141.

46. Shi, J., Zhou, T. & Chen, Q. Exploring the expanding universe of small RNAs. *Nature Cell Biology* **24,** 415–423 (2022).

Methods

Xron is trained using both IVT and IP datasets to obtain better performance. It was first trained on a surrogated IVT dataset and then fine-tuned on IP data. To make efficient finetuning and to avoid overfitting to the all-or-none methylated reads in IVT data when training with the long current signal, we create partially methylated reads using data augmentation, first segmenting the signal and then cross-linking the reads and its corresponding signal in silico. To achieve this, we design a novel nonhomogeneous hidden Markov model (NHMM) that can be trained to conduct signal segmentation in a semi-supervised fashion on modified reads, even when lacking methylation labels. The NHMM is trained using the forward-backward algorithm with its transition matrix conditioned on a canonical basecalled sequence and its alignment, thus giving the maximum a posteriori estimation of the model parameters regarding methylation base. The Viterbi path of the NHMM gives the alignment between the current signal and sequence. Following the signal segmentation process with the NHMM, we prepared a partially methylated dataset through data augmentation, splicing the fully methylated and unmethylated segments. Training on this augmented dataset diminishes the inductive bias of the model on partially methylated reads when training with entirely methylated or nonmethylated reads. We then trained an end-to-end methylation-detection basecaller on the augmented dataset, and it achieved high-accuracy methylation base detection at a single-read resolution. We further improved the basecaller by applying a fine-tuning procedure on IP data with label smoothing to obtain a more accurate basecalling model. Finally, we benchmarked different m6A detection methods on three datasets, including a synthetic IVT dataset, a yeast dataset, and a human HEK293T cell line, demonstrating that Xron yields accurate methylation-aware basecall and generalizes to different species.

NHMM trained using semisupervised learning

We design a hybrid framework to conduct signal segmentation and alignment when methylated bases are present. A homogeneous HMM (we refer to this model as an HMM throughout the remainder of this paper for convenience), as employed in the common Nanopolish preprocessing tool [47], faces challenges when applied to sequences with methylation bases. The absence of ground truth for the methylation states in each basecalled sequence prevents supervised HMM training. However, training the HMM unsupervised, using only signal and reference genome, is difficult due to the high noise contained in nanopore sequencing signals, the long lengths of the electrical signals, and the similar signal levels between certain k-mers and their methylated counterparts. Additionally, totally unsupervised training is not necessary as we already have the canonical basecalled sequence with alignment given by the canonical basecaller and the reference genome. Although

the signals are error-prone in the methylated region, they still provide a general sketch of the sequence. Thus, instead of performing unsupervised learning with the HMM, we develop a semi-supervised training process using an NHMM, where we use the basecalled canonical sequence as a prior when building the transition chain backbone in the NHMM. In contrast with an HMM possessing a homogeneous transition matrix that remains constant over time t, an NHMM possesses a nonhomogeneous transition matrix that depends on the external variables and varies over time t, allowing the use of dynamic control for the transition process. Various NHMMs have been used in meteorology [48] and economics [49, 50] by constructing transition matrices that depend on time-varying covariates, such as seasonality [48] or economic cycle indicators [50]. In our case, the base probabilities along time t predicted by an existing canonical basecaller (a base caller trained to predict only canonical bases) are used as the time covariates of the transition matrix. This approach enables the model to concentrate on the section of the Markov chain guided by the predicted base probability (Fig. 1c), rather than dealing with the entire chain as is required in unsupervised learning using HMM, which is more challenging and error-prone.

NHMM for methylated sequence segmentation and alignment

The NHMM represents the input sequence of raw current signals as $X = (x_1, \ldots, x_T)$ for a given k-mer sequence $Z = (z_1, \ldots, z_T)$ inside a nanopore over the sequencing duration T. Each signal point x_t represents a normalized current value, while z_t is a variable indicating the k-mer at time t. The transition matrix of the NHMM is constrained on the basecalled sequence and its alignment given by the canonical basecaller. More specifically, suppose we are given the base probability matrix $H = (h_1, \ldots, h_T) \in \mathbb{R}^{B \times T}$, where B is the number of bases and h_t^b is the probability of base b at time t, which is obtained from an existing canonical neural network basecaller (Fig. 1a) [51, 52]. From the base probability matrix H, we extract the most probable basecalled sequence $Y = \{y_\tau\}$ and its corresponding alignment $A(t)$ which aligns the signal point time t to sequence index τ, giving $t \to \tau$. After correcting the basecalled sequence with the reference genome, we construct a reference k-mer sequence C by sliding a window of size k (in our case, $k = 5$) across the basecalled sequence, moving one base at a time. Each windowed segment forms a k-mer and is added to the sequence $C = \{c_\tau\}$. From now on, to simplify the notation, we use c_t to denote the corresponding k-mer at time t after transitioning through alignment $c_{A(t)}$. All time offsets of the k-mer sequence reside in the sequence domain, meaning c_{t-1} refers to $c_{A(t)-1}$. Finally, we derived the k-mer transition matrix Ψ from k-mer sequence C; for details, see the next section. Then, the likelihood of observing an electrical signal X

is given by:

$$P(X \mid C) = \sum_{Z} \left[\prod_{t=1}^{T} P(x_t \mid z_t) \prod_{t=1}^{T} P(z_t \mid z_{t-1}, c_{t-\lfloor m/2 \rfloor}, \ldots, c_{t+\lfloor m/2 \rfloor}) \right]. \quad (1)$$

Here, Z is the hidden state representing the underlying k-mer sequence, z_t is the k-mer at time t, and $c_{A(t)}$ is the corrected k-mer representation at time t acquired from the canonical neural network output H (Fig. 1a). T is the maximum time stamp for a given sequence segment. m is the window size for the k-mers to be considered. $P(x \mid z)$ is the emission probability of the signal x given the k-mer z, as modeled by a Gaussian distribution.

Constructing a transition matrix from the base-called sequence and its alignment

We loosely constrain the transition matrix at time t in the nonhomogeneous HMM by using the base prediction output H derived from a canonical basecaller, thereby using the segmentation results provided by the basecaller in an error-tolerant manner (Fig. 1b). By calculating the most probable path from H, we can obtain both the basecalled sequence and the alignment between each base within the most probable path and the sequencing time t. Following this, we correct the basecalled sequence using the reference genome, and we also make appropriate revisions to the alignment to address the deletion or insertion errors in the basecalled sequence. We transform the corrected sequence into a k-mer sequence $C = \{c_t : t = 1, \ldots, T\}$, incorporating the k bases surrounding each base in the basecalled sequence; then, this k-mer sequence is reformatted into transition matrices $\Psi = \{\psi_t : t = 1, \ldots, T\}$ by including at most m transitions, where each ψ_t is the temporal transition matrix at time t. During the process of constructing the k-mer sequence C from H, the basecalled RNA sequence is corrected by aligning it to a reference genome through the following steps:

- For mismatched bases, we replace the bases in the k-mer with the reference bases.
- For insertions/deletions in the base-called sequences that are smaller than five bases, we determine the new signal alignment boundary of the inserted/deleted bases by evenly merging/splitting the signal boundaries of nearby bases; i.e., we redistribute the occupancy of the inserted bases to the nearby bases and allocate occupancy for the deleted bases from the nearby bases.
- We skip the sequence segments with insertions and deletions that are larger than five bases for quality control purposes.

The transition matrix Ψ is then constrained by C, masking out the irrelevant transition paths so that only transition paths that are likely to occur at time t are retained. To more clearly see what these temporal transition matrices stand for, let $\psi_{i,j}^t = \Pr(z_t = i \mid z_{t-1} = j, c_{t-\lfloor m/2 \rfloor}, \ldots, c_{t+\lfloor m/2 \rfloor})$ be the transition probability from k-mer i to k-mer j given constraint k-mers c_i from a time window with a width of at most m, i.e., from $t - \lfloor m/2 \rfloor$ to $t + \lfloor m/2 \rfloor$. At the start and end of sequence, the window size is less than k due to boundary constraints. In comparison with the transition matrix $\phi_{i,j} = P(z_t = i \mid z_{t-1} = j)$ of a homogeneous HMM, the transition matrix now changes over time t:

$$\psi_{i,j}^t = \sum_{t'=t-\lfloor m/2 \rfloor}^{t+\lfloor m/2 \rfloor} e_{c_{t'}} \otimes e_{c_{t'+1}} \odot \phi_{i,j}, \tag{2}$$

where $\otimes$ is the tensor product operation, $\odot$ denotes elementwise multiplication, e_i is a one-hot vector where only the i^{th} element is 1, and $\phi_{i,j}$ is the transition matrix in which $\phi_{i,j} = 1$ if the transition from k-mer i to k-mer j is valid (otherwise, it is 0). For example, AAACT to AACTA is valid, while AAACT to ACTCC is not, as we only allow 1 base step. $\psi_{i,j}^t$ is the k-mer transition matrix from the k-mer sequence described above; it is a binary value matrix indicating the k-mer transition $i \to j$ at time t, where 1 denotes a possible transition and 0 represents an impossible transition.

We construct the transition matrix from m nearby k-mers instead of only the k-mer at time t from k-mer sequence C because the base probability predicted by the canonical basecaller is not exact due to the connectionist temporal classification (CTC) loss used [51, 52] and the insertion/deletion errors in the sequence, nor is it totally correct due to the previously unseen modified bases. Thus, we allow the NHMM to explore the alignment space in two ways. First, at each time point, the transition matrix of the NHMM is restricted to the current transition probability and the m nearby transition probabilities, where m is a hyperparameter (Eq. 2). This is done to make sure that the final alignment output by the NHMM is not too far away from the given the alignment from canonical basecalling but still allows for exploration within the m-base window. Second, the transition path of the underlying Markov chain is broadened to encompass all possible modified counterparts for each k-mer along the path (Fig. 1c). As an example, AACGT is extended to include four alternative k-mers with modified bases, AACGT (the original k-mer), AMCGT, MACGT, and MMCGT, leading to expanded paths. After the transition matrix is constructed for all the time points, the NHMM is then trained using the expectation-maximization (EM) algorithm [53] until it converges (Fig. 2b).

Preparing the training data with data augmentation and read sampling

All-or-none methylated reads exhibit either complete methylation of all adenine (A) bases or none at all, whereas in actual biological samples, methylation typically occurs less frequently and is more sporadically distributed. To prevent the neural network from overfitting to all-or-none methylation reads, we create a training dataset containing partially methylated reads with labels. This is accomplished by dividing the signals from the all-or-none modified reads into smaller segments and subsequently splicing them together. The corresponding sequences are recombined according to their alignment with the signal, as provided by the NHMM. Merging the signals generated from distinct k-mers at their junction points can result in substantial discrepancies between the combined signal and the actual signal obtained from a real sequencing run. To avoid such deviations caused by k-mer mismatches, we ensure that the preceding and succeeding k-mers at the joint sections are identical. For instance, we can merge the signal segments with base-called sequences such as GGM***CGTTC***XXX and XXX***CGTTC***TAG to form GGM***CGTTC***TAG. To achieve this, we define nonmethylatable k-mers as k-mers without adenine (***CGTTC*** in the example). They have the same sequencing signal distributions in both modified and unmodified reads, making them suitable for use as joint anchors. We employ the trained NHMM to decode both the canonical and fully modified reads in the training IVT dataset, using the base probability prediction from the canonical basecaller as described before. The alignment between the sequence and signal is established through a Viterbi path, which assigns each signal point to its corresponding k-mer (Fig. 1d). Each read is subsequently divided into segments at nonmethylatable k-mers. These segments are used to construct a k-mer signal graph, where each node represents an invariant k-mer. Each edge corresponds to a signal segment whose aligned sequence begins and ends at the respective k-mers of the connected nodes (Fig. 1e). We then perform a random walk on the graph, choosing the next edge via an ϵ-greedy sampling strategy with an upper confidence bound (UCB) [54], as used in the multi-armed bandit algorithm, to ensure maximum diversity in the sampling sequence (see Algorithm 1.4 in the supplementary materials).

Data processing

Acquisition and processing of direct RNA sequencing datasets All datasets used in this study are acquired from refs [28, 31, 36, 55]. We obtained both replicates (replicate 1 and 2) from the Epinano synthesized IVT RNA dataset [31] and the only single replicate from the ELIGOS synthesized IVT RNA dataset [28]. Both of these datasets contain fully modified reads and unmodified control reads. We also obtained all the NA12878 IVT RNA reads from the Oxford Nanopore human reference dataset repository: `https:`

//github.com/nanopore-wgs-consortium/NA12878/blob/master/RNA.md [55]. For the yeast dataset, we obtained all three replicates of the wild strain and *ime4*-knockout strain (ime4Δ) [31]. Reads are extracted if mapped to m6A-modified RRACH sites previously identified by antibody immunoprecipitation [37]. For the human HEK293T cell line, we obtained two replicates (replicate 1 and 2) of the wild-type human HEK293T cell [36] to evaluate models. Following a previous study [36], we used the reference transcriptome and its genome annotation provided by SG-NEx project: `https://github.com/GoekeLab/sg-nex-data` [41]. We used the same m6A DRACH sites in the m6Anet paper [36], which were originally identified by m6A-seq and miCLIP experiments [9, 12]. All replicates in the datasets are biological replicates, which are independent biological samples sequenced using the same direct RNA nanopore sequencing protocol. As for synthesized IVT reads, RNA replicates were transcribed from synthesized DNA reads with different sequences. See the sections below for details on replicates used for training and evaluating. All samples were generated using the Nanopore R9.4.1 flow cell, except for the human IVT data, which came from the R9.4 flow cell. The only significant difference between the two flow cells is the slightly improved yield in the R9.4.1.

Canonical basecalling and mapping All reads in the training dataset were basecalled using the Guppy 5.0.11 ONT basecaller [56] and then mapped to the reference genome using minimap2 v2.24 [57] with the settings "`-ax map-ont -uf --secondary=no --MD`". The mapped reads were then transferred to the BAM format using Samtools 1.11.0. A canonical neural network basecaller with the same structure as the CRNN was then trained using the NA12878 IVT reads, and this basecaller was then used to produce the base probability prediction. This canonical basecaller is used as a starting model when we retrain it on the augmented IVT data and subsequently fine-tune it on the yeast data [31].

Training datasets We randomly selected 300,000 canonical (unmodified) read chunks and 300,000 fully-modified read chunks from replicate 1 of each of the two synthesized IVT RNA datasets [28, 31], as well as the first 300,000 canonical read chunks from the Oxford Nanopore Human IVT reference dataset [55] to construct the k-mer signal graph we described above. Reads were filtered out if the corresponding basecalled sequence was shorter than three bases, if the signal had a dwell time (the putative duration a k-mer remains in the pore) exceeding 2000 signal time points, if the basecalled sequence could not be aligned to the reference genome, or if a single base type comprised more than 60% of the basecalled sequence. This filtering process resulted in 228,983 canonical read chunks and 204,822 methylated read chunks from the first synthesized IVT dataset [31], 195,161 canonical read chunks and 213,085 methylated read chunks from the second synthesized IVT dataset [28], and 188,004 canonical read chunks from the Human IVT reference dataset [55]. Methylation

sites identified by antibody immunoprecipitation [37], derived from the first replicate of the wild-type and the first replicate of the ime4Δ from the yeast dataset [31] were used to create the fine-tuning dataset. We regarded all sites from the wild-type strain as methylated and all sites from the ime4Δ strain as unmethylated. However, we considered these classifications noisy labels and used label smoothing during fine-tuning. Human HEK293T cell dataset [36] was not used for training and only used in the evaluation.

Evaluation datasets All the accuracy evaluation datasets we used are sourced from previously published resources. These include a synthesized IVT dataset [31], a yeast dataset [31], and a human HEK293T cell dataset [36]. We used the second replicate from both the synthesized IVT and yeast datasets, as we had already used the first replicate of these two datasets for training and fine-tuning, and we used the first replicate of the human HEK293T cell dataset as it was not included in training. A subset of the human HEK293T cell dataset containing 500 genes was randomly sampled from the original dataset. For the yeast data, we assessed model performance based on the sites identified by m6A-seq [37] for the wild-type strain, and the ime4Δ strains where no methylation should be observed. For evaluation on human data, following previous work [36], we regarded the combined sites identified by m6A-seq [14] and miCLIP [12] as methylated sites, and other randomly selected sites with the DRACH motif as unmethylated sites.

Training and fine-tuning a m6A methylation-sensitive neural network basecaller

We used the partially modified reads sampled from the signal k-mer graph to retrain a canonical basecaller. Before performing retraining on the pre-trained canonical basecaller, we reinitialized the parameters of the last fully connected hidden layer with random weights but kept the same standard deviation. We then retrained the model using a smaller learning rate (0.00001) than the usual learning rate (0.001). We fine-tuned our model on biological samples with m6A sites identified by antibody experiments [31], labeling the A base at each modified site as an m6A base for every read (Fig. 2b). Since the bases at methylation sites are usually not methylated in every read, this approach would introduce many false-positive labels. To address this issue, we applied label-smoothing to the connectionist temporal classification (CTC) loss that was used to train the basecaller. A label sequence of length L was defined as $S = \{s_i : i = 1, 2, \ldots, L\}$, and each s_i belonged to the set $\{A, C, G, T, M\}$. The base probability logit output $H \in \mathrm{R}^{T/K \times N}$ was a (T/K)-by-N matrix derived from the basecaller's CRNN, where K is the total number of strides (i.e., the number of steps the convolutional filter moves across the input at each operation), and N is the number of bases used for prediction plus 1 (a blank symbol). The altered CTC loss with label smoothing under a strength factor

represented by ϵ was then defined as:

$$L = \epsilon L_{CTC}(S_{M\to A}, H) + (1-\epsilon)L_{CTC}(S, H). \tag{3}$$

where M stands for the m6A base, L_{CTC} is the usual CTC loss, and $S_{M\to A}$ is the sequence in which every m6A base is replaced with an A base. We set $\epsilon = 0.1$ empirically for the fine-tuning process, with an expectation that the methylation label is correct with probability $1-\epsilon$.

References

47. Simpson, J. T. *et al.* Detecting DNA cytosine methylation using nanopore sequencing. *Nature Methods* **14,** 407–410 (2017).

48. Hughes, J. P., Guttorp, P. & Charles, S. P. A non-homogeneous hidden Markov model for precipitation occurrence. *Journal of the Royal Statistical Society Series C: Applied Statistics* **48,** 15–30 (1999).

49. Netzer, O., Lattin, J. M. & Srinivasan, V. A hidden Markov model of customer relationship dynamics. *Marketing Science* **27,** 185–204 (2008).

50. Meligkotsidou, L. & Dellaportas, P. Forecasting with non-homogeneous hidden Markov models. *Statistics and Computing* **21,** 439–449 (2011).

51. Graves, A., Fernández, S., Gomez, F. & Schmidhuber, J. *Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks* in *Proceedings of the 23rd International Conference on Machine Learning* (2006), 369–376.

52. Teng, H. *et al.* Chiron: translating nanopore raw signal directly into nucleotide sequence using deep learning. *GigaScience* **7,** giy037 (2018).

53. Baum, L. E., Petrie, T., Soules, G. & Weiss, N. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics* **41,** 164–171 (1970).

54. Sutton, R. S. & Barto, A. G. *Reinforcement learning: An introduction* (MIT press, 2018).

55. Workman, R. E. *et al.* Nanopore native RNA sequencing of a human poly (A) transcriptome. *Nature Methods* **16,** 1297–1305 (2019).

56. Oxford Nanopore Technologies. *Guppy* `https://community.nanoporetech.com/posts/guppy-v5-0-11-patch-releas`. Version 5.0.11. [Online; accessed 23-July-2023]. 2021.

57. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34,** 3094–3100 (2018).

1 Supplementary Text

1.1 Summary of prior approaches

Table S1. Prior approaches concerning the identification of m6A modifications. AON: All-or-none modified dataset. KO: Knockout dataset.

Method	Structure	Training data	Input features
Epinano (2019) [31]	SVM	AON	Basecall error & Signal feature
ELIGOS (2021) [28]	Stat. anal.	–	Basecall error
Nanocompore (2021) [29]	GMM	–	Signal features
nanom6A (2021) [33]	GBM	AON	Signal segments
CHEUI (2022) [35]	CNN	AON	Signal segments
m6Anet (2022) [36]	MLP	In vivo KO	Signal features & Sequence
Xron (This work)	CRNN	AON & In vivo KO	Raw signals

1.2 K-mer encoded as integer

We encoded each k-mer with an integer by initially converting the k-mer string into a base-b integer. For example, 'ACGTM' is represented as a base-5 integer 01234_5. This base-5 integer is then converted into a base-10 integer (z_t), where 01234_5 is transformed to 112_{10}.

1.3 Signal segmentation

To determine the exact alignment between the raw current signals and the corresponding transcription positions, a signal segmentation procedure is typically required to assign consecutive signal points (called an event) to each base pair. The electrical current signals acquired from the ONT sequencer are 1D time-series signals sampled at 4,000 points per second. Under the direct RNA sequencing protocol, the average movement speed of RNA through the pore is 70 base pairs per second, resulting in an average of 57 sampling points per base pair. The signal level and duration of an event are decided by the five nucleotides inside the pore, where the middle nucleotide is the one to which we mapped.

1.4 Sampling algorithm

Algorithm 1 Signal-k-mer Graph Random Walk Sampling

Input:

$G(V, E)$ ▷ Signal-k-mer graph with nodes V and edges E
N ▷ max number of segments to sample
L ▷ max length of each sampled segment
$\epsilon = 0.1$ ▷ exploration when sampling start node
$\gamma = 0.1$ ▷ exploration factor when sampling edge

Output:

S ▷ Sampled reads

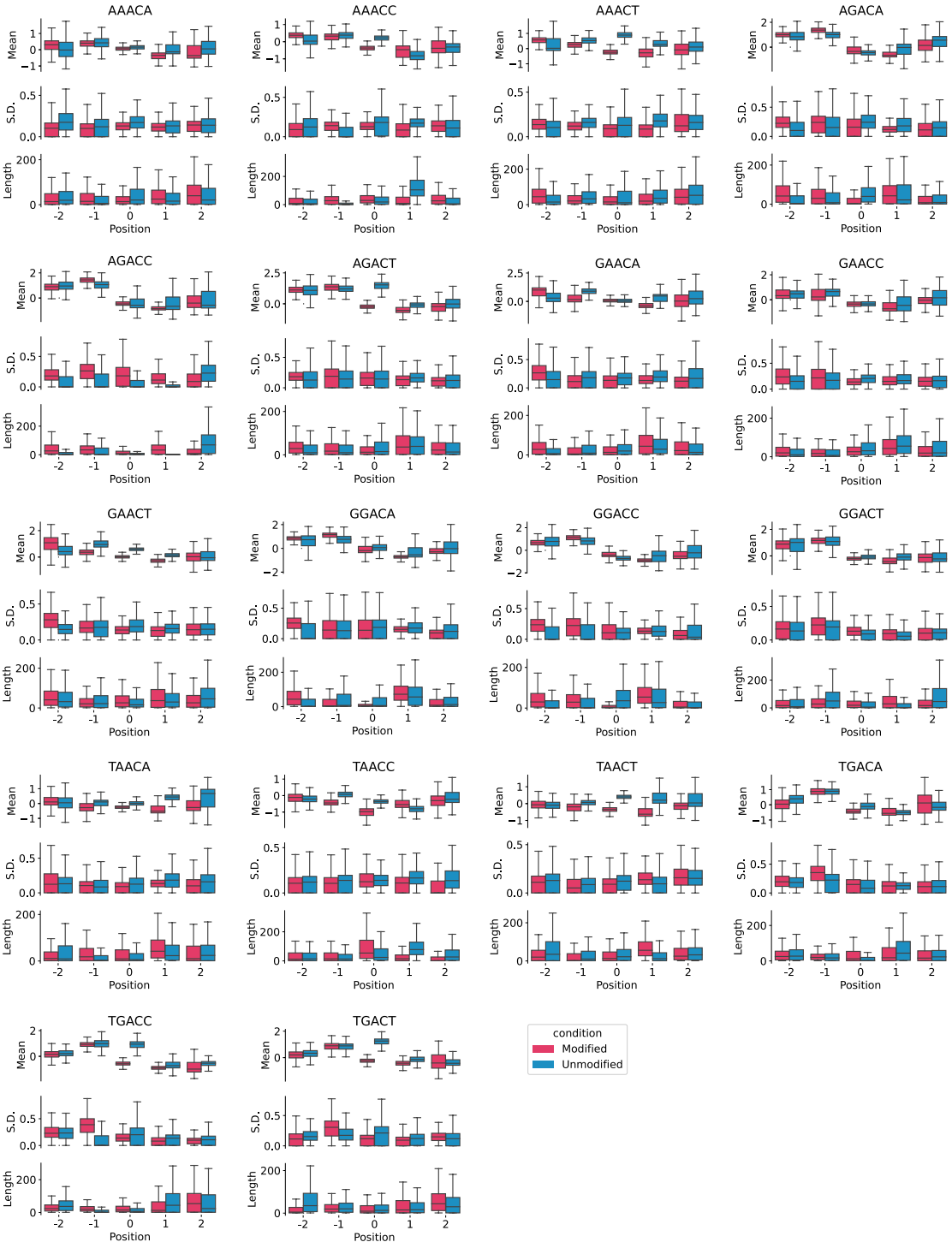
$S \leftarrow []$
v.weights = #edges starting with v, for all $v \in V$
e.visits = 0, for all $e \in E$
while len(S) $< N$ **do**
    curr_s = []

    Pick the start node:
    Generate a random number $r \in [0, 1]$
    if $r < \epsilon$ **then**
        $v \leftarrow$ random node $\in V$
    else
        $v \leftarrow \text{argmax}_x(x.\text{weight}, x \in V)$
    end if

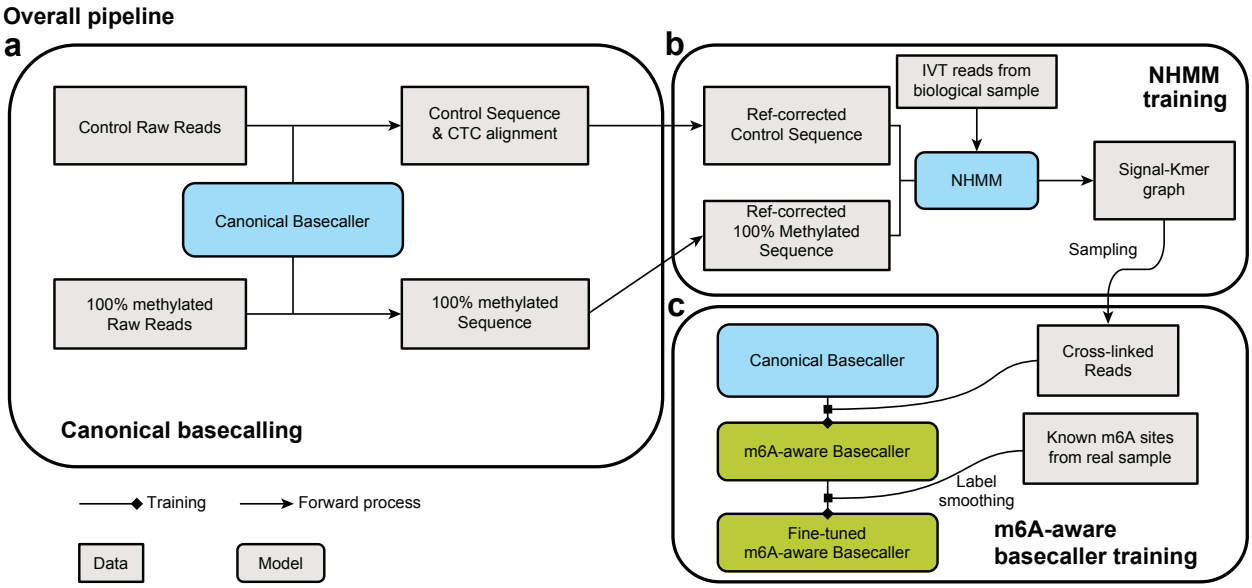
    Random walk along the graph:
    while len(curr_s) $< L$ **do**
        $p = [\sqrt{\text{len}(S)/x.\text{visits}}$ for x in v.edges$] + \alpha * [q(x)$ for x in v.edges$]$ ▷ Upper Confidence Bound
                ▷ $q(x)$ is the entropy of sequence x, v.edges are edges starting from node v
        $p = p/p$.sum()
        Generate a random number $r \in [0, 1]$
        if $r < \gamma$ **then**
            e = random choose e from v.edges
        else
            e = choose e according to p
        end if
        curr_s.append(e)
        e.visits $\leftarrow e$.visits + 1
        curr_v.weights $\leftarrow \#\{v.\text{edges}\}/\sqrt{\text{sum}([x.\text{visits for } x \text{ in } v.\text{edges}])}$
    end while
    S.append(curr_s)
end while

Table S2. Basecalling accuracy comparison between Xron and Guppy on three different datasets and their control datasets. The deletion, insertion, and mismatch rates (%) were calculated as the numbers of deleted, inserted, and mismatched bases divided by the number of bases in the reference sequence, respectively. The identity rate (%) was defined as the number of matched bases in the query sequence divided by the number of bases in the reference sequence (the higher the better). All reported rates are mean values among the aligned reads.

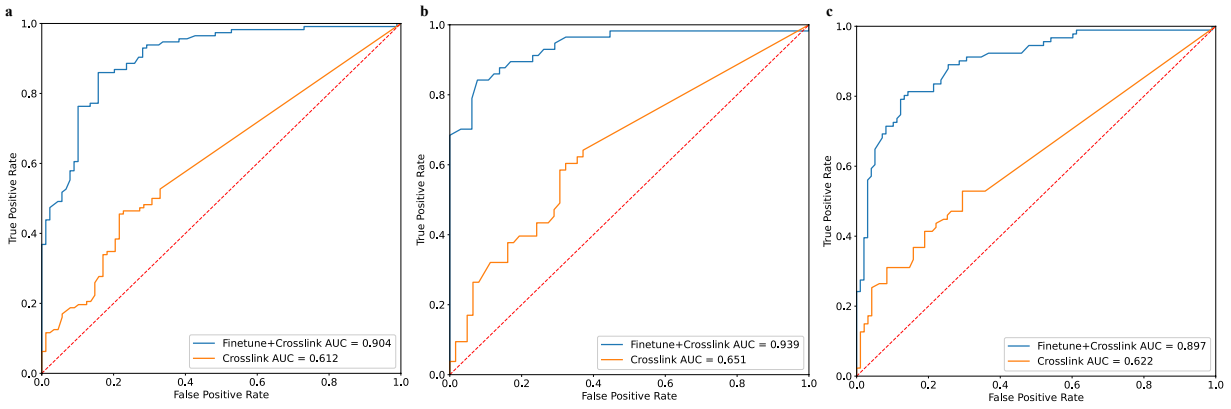
Condition	**Model**	Deletion rate (%)	Insertion rate (%)	Mismatch rate (%)	Identity rate (%)(↑)
IVT Control	Xron	4.14	11.60	8.51	87.35
	Guppy	4.30	2.20	2.95	92.75
IVT m6A	Xron	5.09	15.04	6.44	88.48
	Guppy	9.11	4.45	12.60	78.28
Yeast ime4Δ KO	Xron	9.47	4.54	5.57	84.97
	Guppy	4.97	2.80	2.54	92.50
Yeast	Xron	9.12	3.83	6.92	83.96
	Guppy	4.80	2.38	3.26	91.94
HEK293T Mettl3 KO	Xron	10.41	1.91	3.68	85.91
	Guppy	4.42	2.59	2.39	93.19
HEK293T	Xron	9.46	2.08	3.43	87.12
	Guppy	11.31	2.45	3.05	85.64



Supplementary Figure 1. Signal features comparison for all DRACH motifs between modified and unmodified sites extracted from the Epinano IVT dataset. Box plot comparing the distributions of mean, standard deviation, and length between modified and unmodified sites for all 18 DRACH motifs. Horizontal lines show the median, the box denotes the interquartile range, and the whiskers extend to 1.5 times the interquartile range. Points beyond this range are considered outliers and are removed from the plot.



Supplementary Figure 2. Overall training pipeline of Xron training. **a** Basecalling the modified and unmodified reads using a canonical basecaller. **b** Training the NHMM with the corrected synthesized RNA sequence and IVT reads from human reference data. The trained NHMM was used to generate a signal k-mer graph. **c** The Xron m6A-distinguishing Basecaller was trained using the cross-linked reads sampled from the signal k-mer graph and then fine-tuned on the yeast and human datasets, where putative m6A sites were identified through an immunoprecipitation experiment. We applied label smoothing when fine-tuning the model due to the noisy m6A labels, as the m6A modification for each read was unknown.



Supplementary Figure 3. Ablation study of Xron model. To validate the necessity of finetuning Xron on IP data, an ablation study was conducted. We evaluate the performance of Xron on three biological replicates (a-c) of yeast data, with and without IP data finetuning. The plots show a dramatic decrease in model performance without finetuning using IP data. Xron model was finetuned using the first replicate of the yeast data.