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) (Liu et al. 2019)	SVM	AON	Basecall error & Signal feature
ELIGOS (2021) (Jenjaroenpun et al. 2021)	Stat. anal.	–	Basecall error
Nanocompore (2021) (Leger et al. 2021)	GMM	–	Signal features
nanom6A (2021) (Gao et al. 2021)	GBM	AON	Signal segments
CHEUI (2022) (Mateos et al. 2022)	CNN	AON	Signal segments
m6Anet (2022) (Hendra et al. 2022)	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.

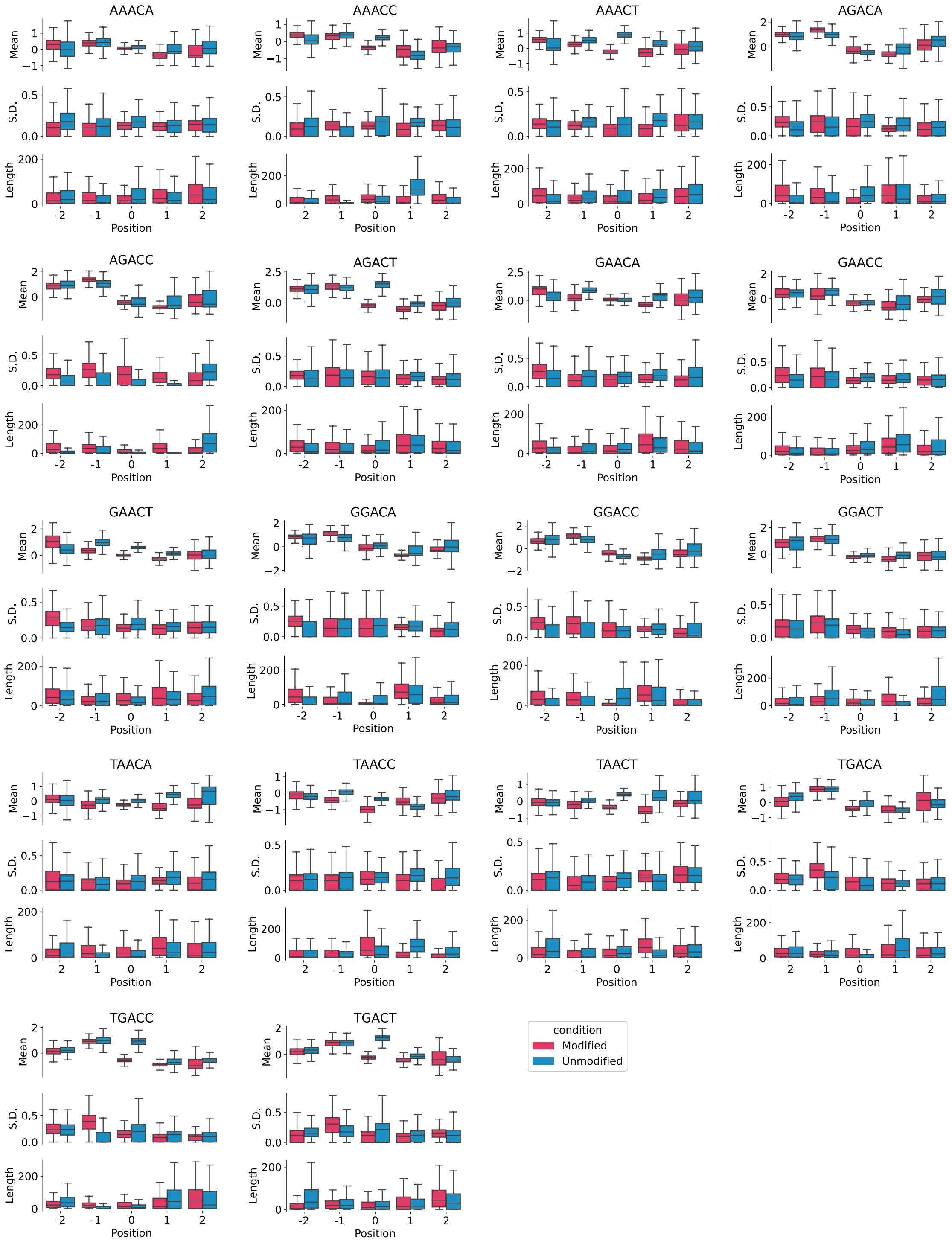
1.4 Sampling algorithm

Algorithm 1 Signal-*k*-mer Graph Random Walk Sampling

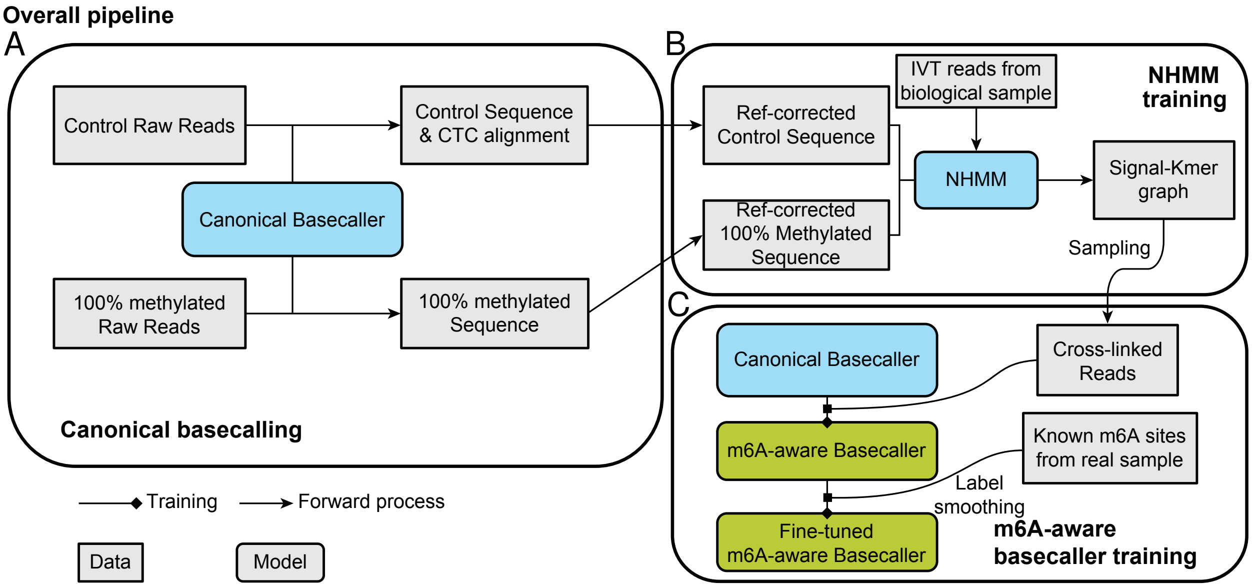
```
Input:
    $G(V, E)$                    ▷ Signal-k-mer graph with nodes $V$ and edges $E$
    $N$                          ▷ max number of segments to sample
    $L$                          ▷ max length of each sampled segment
    $\epsilon = 0.1$             ▷ exploration when sampling start node
    $\gamma = 0.1$               ▷ 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
        $v \leftarrow$ argmax$_x(x$.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 $\leftarrow$ #$\{v$.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.

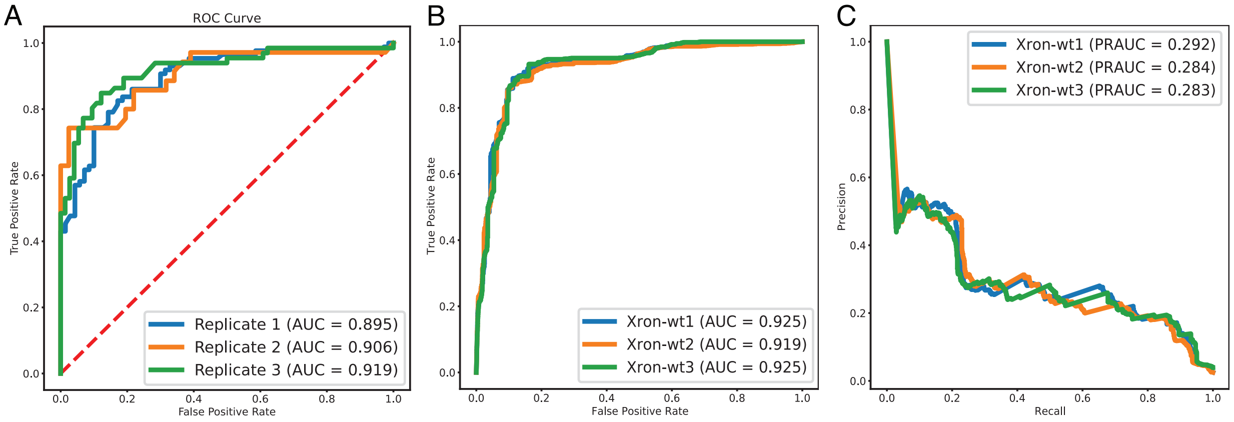
Condition	**Model**	Deletion rate (%)	Insertion rate (%)	Mismatch rate (%)	Identity rate (%)(↑)
IVT Control	Xron	4.14	11.60	8.51	87.35
	Guppy	4.30	2.20	2.95	92.75
IVT m6A	Xron	5.09	15.04	6.44	88.48
	Guppy	9.11	4.45	12.60	78.28
Yeast *ime4Δ* KO	Xron	9.47	4.54	5.57	84.97
	Guppy	4.97	2.80	2.54	92.50
Yeast	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



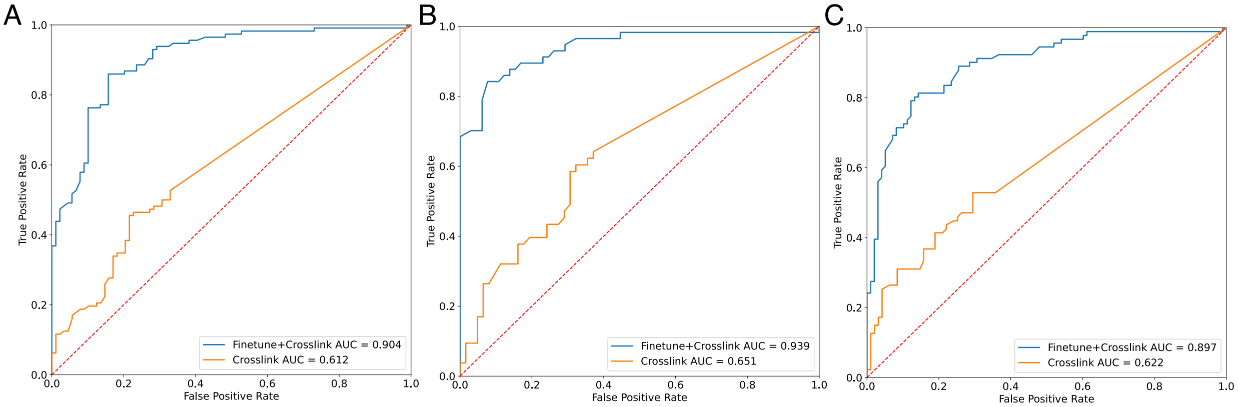
Supplementary Figure S1. Signal features comparison for all DRACH motifs between modified and unmodified sites extracted from the Epinano IVT dataset. Box plot comparing the distributions of mean, standard deviation, and length between modified and unmodified sites for all 18 DRACH motifs. Horizontal 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.


Overall pipeline
A
Control Raw Reads
Control Sequence & CTC alignment
Canonical Basecaller
100% methylated Raw Reads
100% methylated Sequence
Canonical basecalling
B
IVT reads from biological sample
NHMM training
Ref-corrected Control Sequence
NHMM
Signal-Kmer graph
Ref-corrected 100% Methylated Sequence
Sampling
C
Canonical Basecaller
Cross-linked Reads
m6A-aware Basecaller
Known m6A sites from real sample
Label smoothing
Fine-tuned m6A-aware Basecaller
m6A-aware basecaller training
Training
Forward process
Data
Model

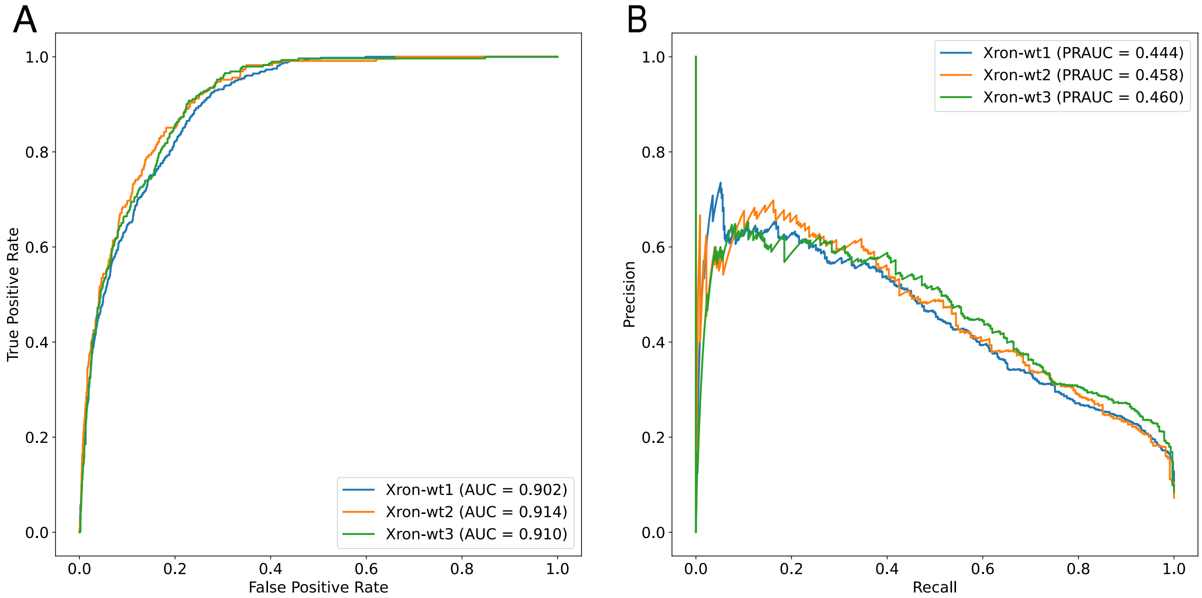
Supplementary Figure S2. Overall training pipeline of Xron training. (*A*) Basecalling the modified and unmodified reads using a canonical basecaller. (*B*) Training the NHMM with the corrected synthesized RNA sequence and IVT reads from human reference data. The trained NHMM was used to generate a signal *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.



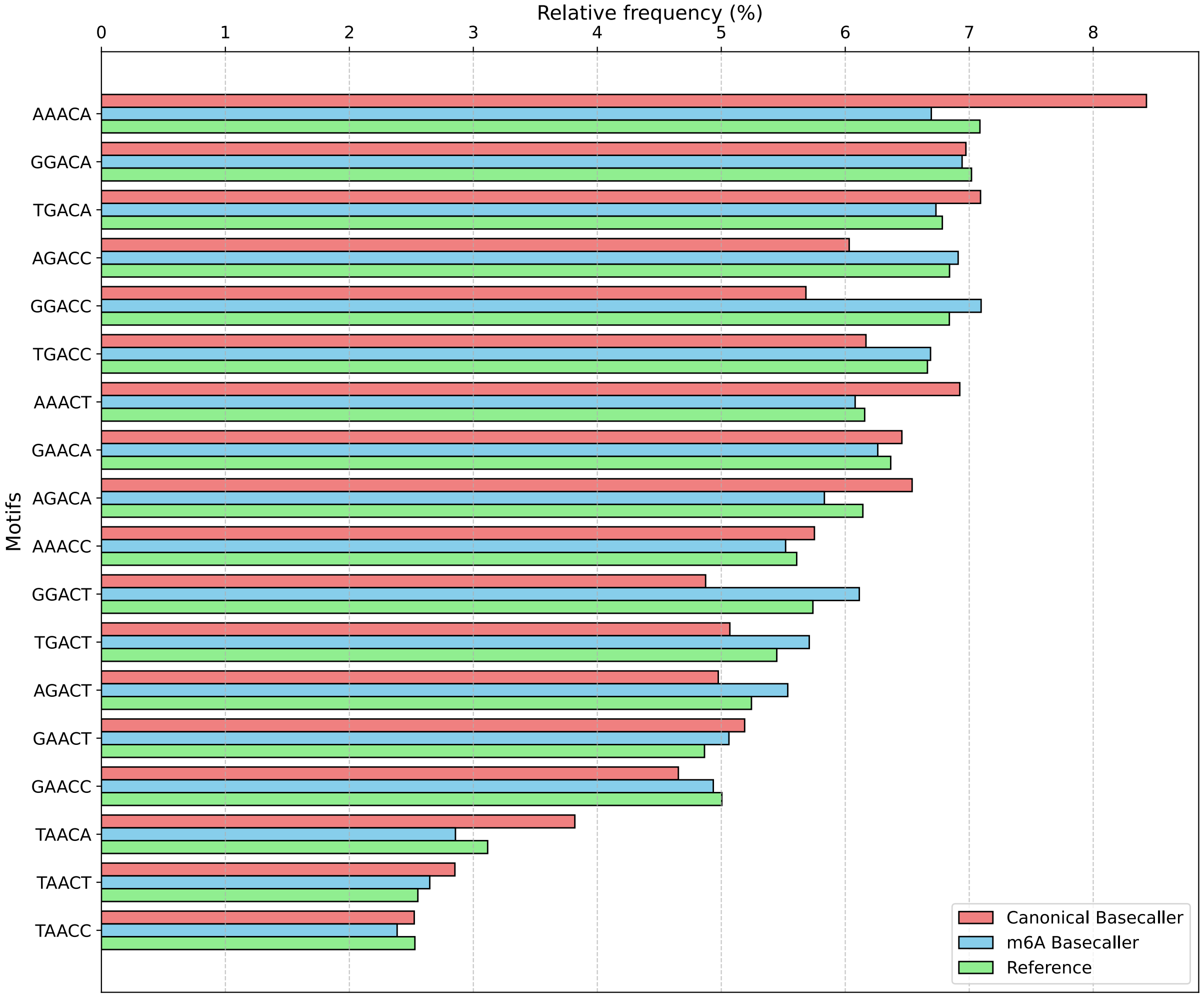
Supplementary Figure S3. Model trained on HEK293T cell line data and evaluated on yeast and Arabidopsis datasets. A model is fine-tuned using human HEK293T cell line data, and then evaluated on the (*A*) yeast *ime4* KO dataset using ROC-AUC, and on the (*B*) Arabidopsis datasets using ROC-AUC and (*C*) PRAUC. The model has a similar performance compared to those fine-tuned on the yeast dataset.



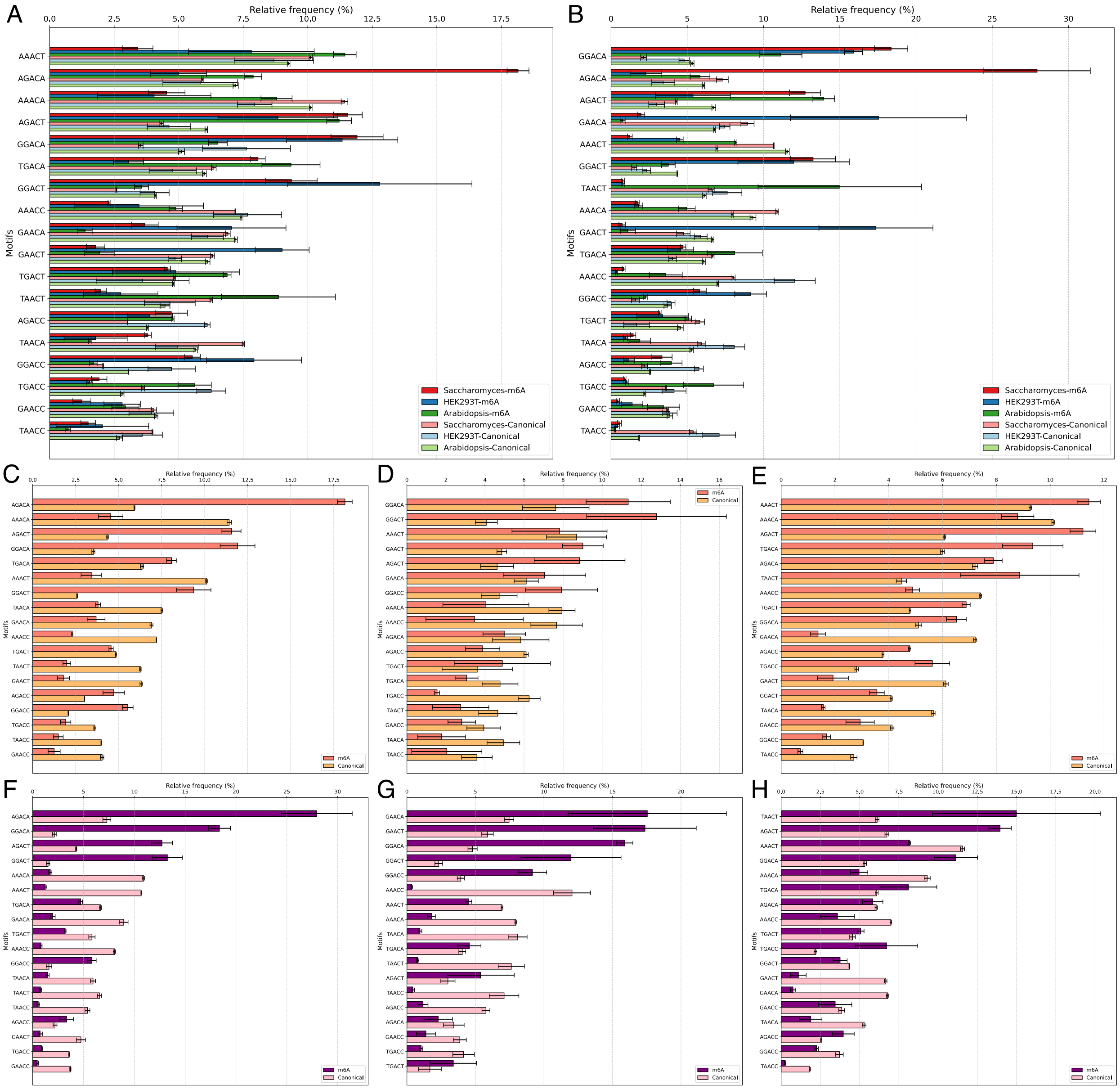
Supplementary Figure S4. Ablation study of Xron model. To validate the necessity of finetuning Xron on IP data, an ablation study was conducted. We evaluate the performance of Xron on three biological replicates (*A*-*C*) of yeast data, with and without IP data finetuning. The plots show a dramatic decrease in model performance without finetuning using IP data. Xron model was finetuned using the first replicate of the yeast data.



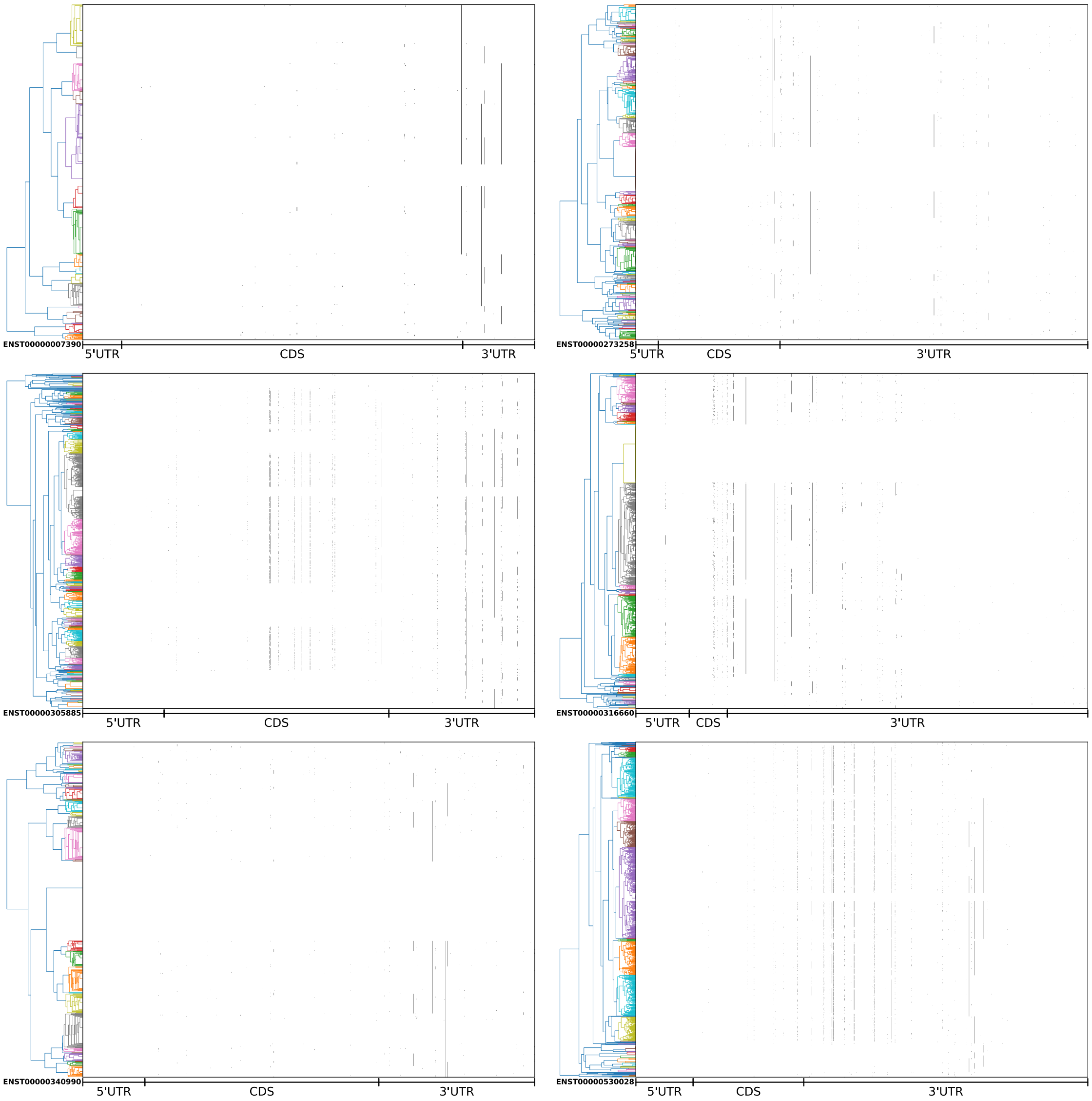
Supplementary Figure S5. Evaluation on all three replicates of Xron model on human HEK293T cell line. The AUC (*A*) and PR (*B*) curve of the Xron model, which is fine-tuned on the first replicate of the yeast dataset, are based on all three replicates of the HEK293T cell line.



Supplementary Figure S6. K-mer frequency comparison between the canonical basecaller and m6A-aware basecaller. To check for any potential context bias in the basecaller, we examined the *k*-mer frequency of the basecalled sequences on the HEK293T cell line, where the canonical basecaller exhibits the most significant performance drop. The results show that, compared to the m6A-aware basecaller, the canonical basecaller has a more deviated *k*-mer distribution in several *k*-mers from DRACH motifs. This deviation indicates a reason for the lower identity rate when basecalling these m6A-modified reads using a canonical basecaller.



Supplementary Figure S7. K-mer frequency for modified and unmodified sites. K-mer frequency of DRACH motifs is analyzed among the Saccharomyces (yeast), HEK293T cell line, and Arabidopsis datasets. Bar plot comparing the proportion of methylated/unmethylated motifs counted for every site (*A*) or every read (*B*), where n reads aligned on the same site count as 1 in (*A*) and as n in (*B*). The bar plot center gives the mean frequency of the 18 fivemer DRACH motif among the 3 replicates while the error bar represents the 95% confidence interval. Separate plots for different datasets (Saccharomyces (yeast), HEK293T cell line, and Arabidopsis) were given for each site (*C* - *E*), and each read (*F* - *H*) to make a clear comparison the frequency between the methylated and canonical motifs. All reads are basecalled using Xron basecaller.



Supplementary Figure S8. Clustering of modification states Clustering plot of 6 genes with multiple modification sites. Clustering was conducted using the SciPy hierarchical linkage module with Yule distance. Asynchronous modification is observed near the end of the CDS and in the $3'$ UTR region in all 6 transcripts (ENST00000340990, $n = 921$; ENST00000305885, $n = 997$; ENST00000530028, $n = 1201$; ENST00000007390, $n = 780$; ENST00000316660, $n = 1296$; ENST00000273258, $n = 1062$).

References

Baum LE, Petrie T, Soules G & Weiss N. 1970. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *Ann Math Stat* **41**: 164–171. doi: 10.1214/aoms/1177697196

Graves A, Fernández S, Gomez F & Schmidhuber J (2006). “Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks”. In: *Proceedings of the 23rd International Conference on Machine Learning*, 369–376. doi: 10.1145/1143844.1143891.

Hughes JP, Guttorp P & Charles SP. 1999. A Non-Homogeneous Hidden Markov Model for Precipitation Occurrence. *J Roy Statistical Society* **48**: 15–30. doi: 10.1111/1467-9876.00136

Li H. 2018. Minimap2: Pairwise Alignment for Nucleotide Sequences. *Bioinformatics* **34**: 3094–3100. doi: 10.1093/bioinformatics/bty191

Meligkotsidou L & Dellaportas P. 2011. Forecasting with Non-Homogeneous Hidden Markov Models. *Statist Comput* **21**: 439–449. doi: 10.1007/s11222-010-9180-5

Netzer O, Lattin JM & Srinivasan V. 2008. A Hidden Markov Model of Customer Relationship Dynamics. *Marketing Sci* **27**: 185–204. doi: 10.1287/mksc.1070.0294

Oxford Nanopore Technologies (2021). *Guppy*. Version 5.0.11.

Simpson JT, Workman RE, Zuzarte P, David M, Dursi LJ & Timp W. 2017. Detecting DNA Cytosine Methylation Using Nanopore Sequencing. *Nat Methods* **14**: 407–410. doi: 10.1038/nmeth.4184

Sutton RS & Barto AG. 2018. In *Reinforcement Learning: An Introduction*. MIT press, Cambridge, Massachusetts.

Teng H, Cao MD, Hall MB, Duarte T, Wang S & Coin LJ. 2018. Chiron: Translating Nanopore Raw Signal Directly into Nucleotide Sequence Using Deep Learning. *Gigascience* **7**: giy037. doi: 10.1093/gigascience/giy037

Workman RE, Tang AD, Tang PS, Jain M, Tyson JR, Razaghi R, Zuzarte PC, Gilpatrick T, Payne A, Quick J, et al. 2019. Nanopore Native RNA Sequencing of a Human Poly (A) Transcriptome. *Nat Methods* **16**: 1297–1305. doi: 10.1038/s41592-019-0617-2