AROWANA: A TRANSFORMER-BASED BASECALLER AND RNA MODIFICATION-CALLER TRAINING FRAMEWORK

Yuk Kei Wan[1,2,5,*], Christopher Hendra[1,5], Bing Shao Chia[1], Wei Leong Chew[1,3], & Jonathan Göke[1,4,*]
[1] Genome Institute of Singapore (GIS), Agency for Science, Technology and Research (A*STAR), Singapore
[2] Department of Medicine, Yong Loo Lin School of Medicine, National University of Singapore, Singapore
[3] Synthetic Biology Translational Research Programme,Yong Loo Lin School of Medicine,
National University of Singapore, Singapore
[4] Department of Statistics and Data Science, National University of Singapore, Singapore
[5] Contributed equally
[*] Corresponding authors (wanyk@gis.a-star.edu.sg and gokej@gis.a-star.edu.sg)

ABSTRACT

Machine-learning methods have enabled RNA modification detection from nanopore direct RNA sequencing. However, the existing nanopore-based RNA modification detection tools are limited, as each modification model requires a large amount of data and compute resources for training. Here we developed arowana, a transformer-based training framework for basecaller and RNA modification detection. We trained arowana modification callers and showed their ability to detect nine modifications stemming from the four nucleotide bases accurately. This demonstrates arowana's potential to be expanded to other modifications.

1 INTRODUCTION

Naturally occurring RNA modifications are chemically modified nucleotide bases from co- and post-transcriptional processing. These RNA modifications including pseudouridine (ψ), 5-methylcytidine (m5C), and N6-methyladenosine (m6A) affect various biological processes, some of which are associated with diseases including cancers (Roundtree et al. (2017b); Hong et al. (2022); Song et al. (2022); Roundtree et al. (2017a)). Apart from these naturally existing ones, RNA modifications are vital to commercial RNA products, including therapeutics, composed of synthetic RNAs replacing unmodified uracil with N1-methyl-pseudouridine (m1ψ) or 5-methoxyuridine (mo5U) to lower the products' immunogenicity, ensuring product efficacy and safety (Karikó et al. (2008); Morais et al. (2021)).

Conventionally, RNA modifications are measured by mass spectrometry, but mostly in rRNAs and tRNAs due to its unscalable sample preparation requirement. Short-read-sequencing-based epitranscriptomic profiling methods have widened RNA modification detection in various organisms, enabling the generation of epitranscriptomic maps. Still, the convoluted wet-lab procedures required by these methods limit RNA modification detection adoption across laboratories. Nanopore direct RNA sequencing has made RNA modification detection accessible, where neural-network-based modification callers differentiate the nanopore-emitted current signals from the translocation of modified and unmodified nucleotides with high accuracy (Zhang et al. (2022); Wan et al. (2022); Zhao et al. (2022); Jain et al. (2022); Begik et al. (2022)). Still, these models require a large amount of data and compute resources for training; hence, existing models for the latest direct RNA-sequencing RNA004 chemistry only limit to detecting m6A, m5C, Inosine, and ψ (ONT-Dorado; ONT-Remora).

The transformer architecture has shown success in speech-to-text translation, which is similar to basecalling nanopore current signals to their corresponding nucleotide bases. Moreover, the transformer decoder representations learn rich information from model training and can serve as transferable features for downstream machine-learning tasks (Radford et al. (2023); Chemudupati et al. (2023)). Adapting OpenAI's Whisper speech-to-text transformer model for nanopore basecalling,

we developed arowana, a direct RNA sequencing-specific basecaller and modification-caller training framework, which trains logistic regression classifiers with the transformer decoder representations (Radford et al. (2023); Chemudupati et al. (2023)). We trained arowana modification callers from in vitro-transcription-generated modified samples and showed that the arowana modification callers can accurately detect nine modifications from the four nucleotides (A, C, G, T) at a single-molecule level. arowana demonstrates that decoder representations serve well as informative features and can achieve high accuracy even with a simple model. This eliminates the need to train complex models for modification detection, showing arowana's potential to train modification callers for any RNA modifications of interest.

2 METHODS

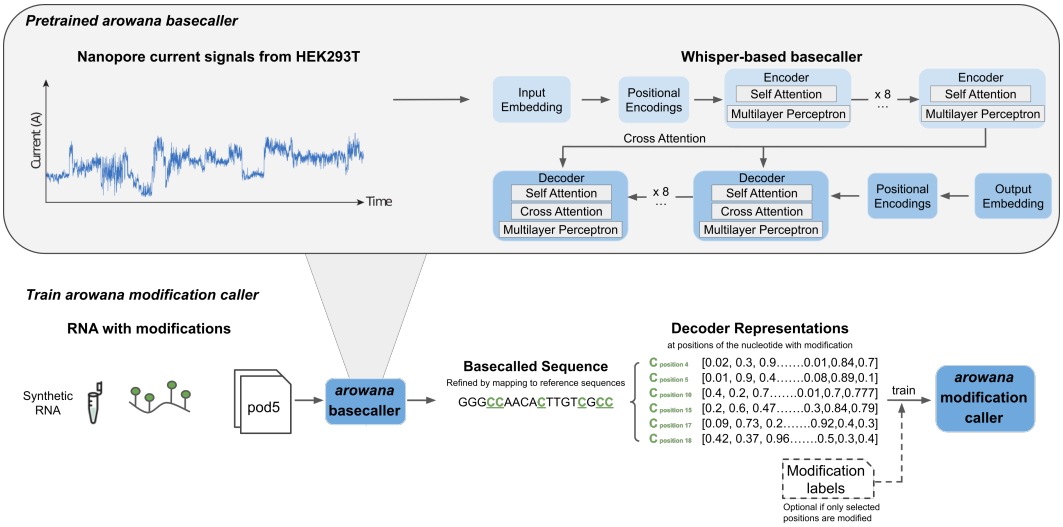


Figure 1: The arowana basecaller and modification caller training framework.

2.1 PRETRAINING THE AROWANA BASECALLER

2.1.1 TRAINING DATA

The arowana basecaller was trained using HEK293T RNA004 direct RNA sequencing data from the Singapore Nanopore Expression Project (SGNEx) (Chen et al. (2021)), which includes pod5, fastq, and bam files. We use 10 randomly selected signal and sequence segments from each of the 13,500 genes with more than 10 signal and sequence segments for training the arowana basecaller.

2.1.2 DATA PROCESSING FOR MODEL TRAINING

The pod5 files were merged into one pod5 file with the pod5 merge command. The pod5 files were converted to blow5 files with blue-crab version v0.2.0 with the p2s command. Segmentation was performed with f5c version 1.2. The reads from the fastq file and the raw signals from blow5 file were indexed with the f5c index command. Then, f5c eventalign was run with the options –rna –min-mapq 0 –min-recalib-events 100 –signal-index –scale-event –print-read-names, and the –kmer_model was used to specify the RNA004 kmer model. The output eventalign.txt file from f5c eventalign was then processed with the arowana basecall_train_prep command, where each read is separated into segments with 3,000 or fewer signal units and 150 or fewer nucleotides. The output of arowana basecall_train_prep contains the following information for training the arowana basecaller: read id, transcript id, the start and end of the signal, and the start and end positions of the sequence based on the reference sequence.

2.1.3 MODEL ARCHITECTURE

We adapted OpenAI's open-source Whisper speech-to-text model, where we use eight encoder layers and eight decoder layers, to convert direct-RNA-sequencing signals to nucleotides [14]. The arowana-adapted Whisper model takes in 3,000 signal unit segments to predict each nucleotide of each sequence sequentially with beam search based on five tokens corresponding to the nucleotides (A, C, G, T) and the end of sequence token (E). The predicted segments of each read are merged and aligned to the reference sequences with mappy version 2.28.

We also adapted Whisper to additionally output the decoder's representations, which are in the dimensions of NR = 150 $\times$ 256, with 256 features corresponding to each of the 150 nucleotide positions (Radford et al. (2023)). As some sequence segments are shorter than 150 nucleotides, we trim the representations corresponding to the length of the sequence segment and merge the trimmed representations from all sequence segments from each read. With parasail version 1.3.4, we perform pairwise alignment of each basecalled sequence to their mappy-aligned reference sequence and assign the representations based on the reference nucleotide.

2.1.4 MODEL TRAINING

We trained the arowana basecaller for 30 epochs with 1 NVIDIA® RTX™ 5000 Ada 16GB GDDR6 GPU for one week. The following hyperparameters were used for training: a learning rate of 9e-5, an effective batch size of 62, and total accumulation steps of 32.

2.1.5 MODEL DISTILLING

We distilled the arowana basecaller by freezing the weights of the encoder layers and removing the intermediate decoder layers, updating the weights of the first and last decoder layers (Supplementary Figure 1). We trained the distilled arowana basecaller using all reads from the HEK293T RNA004 direct RNA sequencing data from the Singapore Nanopore Expression Project (SGNEx) for five epochs with 1 NVIDIA® RTX™ 5000 Ada 16GB GDDR6 for 18 days. The following hyperparameters were used for training: a learning rate of 9e-5, an effective batch size of 400, and total accumulation steps of 32.

2.2 TRAINING THE AROWANA MODIFICATION-CALLER

2.2.1 TRAINING DATA

Training data consists of direct RNA-Seq data generated (1) using unmodified RNA for the RNA sequence of interest, and (2) using modified RNA for the RNA sequence and the modification of interest. We designed two synthetic sequence libraries with two strategies: sequences containing all possible kmers and sequences representing human cDNAs, and we generated data for nine RNA modifications of interest (m1A, m6A, f5C, m5C, thG, ψ, m1ψ, mo5U, and s2U). Both DNA libraries were pooled in equimolar concentrations and transcribed in vitro with modified and unmodified nucleotides. For the unmodified RNA, only unmodified nucleotides were incorporated during in vitro transcription. For the modified RNAs, one out of the four nucleotides (i.e. one of the A, U, C, or G nucleotide identities) was substituted with modified nucleotides while the other three nucleotides remained unmodified. The transcribed RNAs were then subjected to nanopore direct RNA sequencing via Oxford Nanopore Technology nanopore sequencing, and the resulting pod5 files were directly inputted to the arowana basecaller, which outputs the basecalled sequence and decoder representations for each read.

2.2.2 MODEL ARCHITECTURE

Previous studies have shown that models trained from a diverse distribution of data extract informative representations (Chemudupati et al. (2023); Baevski et al. (2020); Hsu et al. (2021); Huang et al. (2022); Chen et al. (2022)). The authors from a study (Chemudupati et al. (2023)) showed that passing the weighted sum of Whisper's decoder representations to downstream layers allows accurate prediction in downstream machine-learning tasks.

Here, similarly, the arowana basecaller has learned from a diverse distribution of sequences from 13,500 human genes and is generalisable to basecall synthetic sequences not seen in the human

transcriptome. Hence, the arowana basecaller should extract informative representations that are transferable to our downstream machine-learning task of predicting the presence of a modification, even with a simple logistic regression model.

From the arowana basecaller, we obtain the reference-sequence-refined basecalled sequence and the decoder representations corresponding to each nucleotide. For each modification, we use the decoder representations corresponding to the specific nucleotide (A, C, G, or T) as features, where each nucleotide has 256 features. We directly utilise the 256 decoder-representation features to train logistic-regression-based modification detection models with linear_model.LogisticRegression from scikit-learn version 1.4.0.

2.2.3 Model Training and Evaluation

Before model training and evaluation, we split each sample into five sets, whose sequence contexts do not overlap. With these five sets, we performed a 5-fold cross-validation of the modification models. For each round, we used four sets for training and held out the remaining set for testing.

For each arowana modification caller, representations of sites from the unmodified sample were labeled as unmodified ($y_{i,j} = 0$), and representations of sites from the modified samples with the modified base were labeled as modified ($y_{i,j} = 1$). We merged the representations of sites from the unmodified sample and the modified samples for each nucleotide for model training.

We trained logistic-regression-based modification callers for m1A, m6A, f5C, m5C, thG, ψ, m1ψ, mo5U, and s2U with the representations from the unmodified sample and each modified sample. For all training, we set scikit-learn.linear_model.LogisticRegression random_state parameter at 0.

3 Experiments

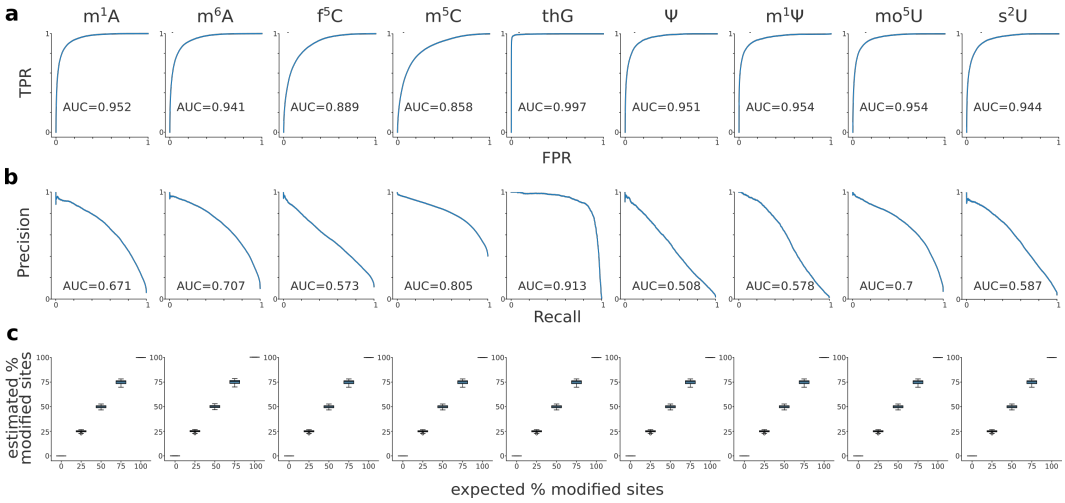


Figure 2: Evaluation of arowana's modification callers' performance. a. Shown are the ROC curves and AUC of the read-level predicted probability modified across nine modifications. b. Shown is the modification ratio estimation across nine modifications based on arowana's predicted modified probability.

3.1 Evaluating the arowana's modification calling models

We evaluated arowana's modification callers by splitting each direct RNA-sequencing sample into independent training and test data (See Methods) to ensure that the sequence context from the training and test sets are mutually exclusive. We ran the arowana mod_inference command on the test set using the model trained with the training set and evaluated the training-set-trained models' ability to distinguish between unmodified and modified reads with the python packages scikit-learn and matplotlib to calculate the ROC curves and the Area under the ROC curves.

Across all modifications, the arowana modification classifiers achieve an average accuracy of ROC AUC 0.895 when classifying unmodified and modified sites (Figure 2a), showing arowana modification classifiers allow accurate single-molecule detection of nine modifications across all nucleotide bases.

3.2 ESTIMATING MODIFICATION RATIOS

We estimated the modification ratios for samples from the test set with the arowana modification callers. We randomly picked 1,000 sites from the unmodified and modified samples at different modified-to-unmodified ratios (0:100, 25;75, 50:50, 75:25, and 100:0) and ran the arowana mod_inference command on each simulated mixture.

To calculate modification ratios for each modification, we used the 90th percentile predicted modified probability from the modified sample as a threshold and calculated the fraction of sites with modified probability above the threshold, which we denote as the modification ratio. We then normalised the modification ratios to be relative to the 100:0 and the 0:100 modified-to-unmodified ratios with the following equation:

$$p_{site} = \frac{p_{site} - min(p_{all})}{max(p_{all}) - min(p_{all})} \tag{1}$$

We obtained the modification ratios from ten rounds of sampling and inferencing and used the python package seaborn to plot the box plots across the samples.

Across the nine modifications, the normalised modification ratios closely resemble the expected proportion from the mixtures with 0%, 25%, 50%, 75%, and 100% modified sites (Figure 2b). This shows that arowana modification callers can accurately estimate the proportion of modified sites in samples with various modifications.

4 DISCUSSION

In summary, we developed arowana, a transformer-based, direct-RNA-sequencing-specific basecaller, and logistic-regression-based modification-caller training framework utilising the informative transformer decoder representations as features. We trained modification callers for nine modifications from all four bases and showed their highly accurate single-molecule modification prediction and modified site proportion estimation. Together, our work shows that the decoder representations from the arowana basecaller serve as informative features for modification calling even with a simple model, hence, removing the requirement of training data- and compute-resource-intensive models for accurate RNA modification detection. Still, the current version of arowana is only evaluated on synthetic sequences not natural transcriptome, so additional evaluation has to be done to confirm arowana's applicability in natural transcriptome.

As we have shown arowana modification callers' success in detecting nine modifications, the pretrained models for the nine modifications are available to users through the arowana mod_inference command. Furthermore, we provide the arowana mod_train command for users to train arowana modification callers for any modifications. Through this command, users can directly provide raw signals from direct RNA-sequencing runs in pod5 format to train arowana modification callers for any modifications of interest. We believe that arowana will open up the profiling of more RNA modifications.

5 ACKNOWLEDGEMENT

Y.K.W. is supported by funding from the Singapore International Graduate Award from Agency for Science, Technology and Research (A*STAR), Singapore. J.G. is supported by funding from A*STAR, Singapore, and by the Singapore Ministry of Health's National Medical Research Council under its Individual Research Grant funding scheme.

6 AUTHOR CONTRIBUTIONS

Y.K.W. and C.H. designed the model and developed the method. B.S.C performed the modified RNA experiments and generated the IVT data with supervision from W.L.C.. Y.K.W. processed the data, trained and evaluated all models, developed the software, and wrote the documentation. Y.K.W. wrote the manuscript with input from C.H. and J.G..

7 CODE AVAILABILITY

arowana is available through GitHub: https://github.com/GoekeLab/arowana

REFERENCES

A Baevski, Y Zhou, A Mohamed, and M Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

O Begik, JS Mattick, and EM Novoa. Exploring the epitranscriptome by native rna sequencing. *RNA*, 28(11), 2022. doi: 1430-1439.

V Chemudupati, M Tahaei, H Guimaraes, A Pimentel, A Avila, M Rezagholizadeh, and T ... Falk. On the transferability of whisper-based representations for" in-the-wild" cross-task downstream speech applications. *arXiv*, 2023.

S Chen, C Wang, Z Chen, Y Wu, S Liu, Z ... Chen, and F Wei. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6), 2022. doi: 1505-1518.

Y Chen, NM Davidson, YK Wan, H Patel, F Yao, HM Low, and ... SG-NEx consortium. A systematic benchmark of nanopore long read rna sequencing for transcript level analysis in human cell lines. *BioRxiv*, 2021.

J Hong, K Xu, and JH Lee. Biological roles of the rna m6a modification and its implications in cancer. *Exp Mol Med*, 54(11):1822–1832, 2022.

WN Hsu, B Bolte, YHH Tsai, K Lakhotia, R Salakhutdinov, and A Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460, 2021.

W Huang, Z Zhang, YT Yeung, X Jiang, and Q Liu. Spiral: Self-supervised perturbation-invariant representation learning for speech pre-training. *arXiv*, 2022.

M Jain, R Abu-Shumays, HE Olsen, and M Akeson. Advances in nanopore direct rna sequencing. *Nat Methods*, 19(10):1160–1164, 2022.

K Karikó, H Muramatsu, FA Welsh, J Ludwig, H Kato, S Akira, and D Weissman. Incorporation of pseudouridine into mrna yields superior nonimmunogenic vector with increased translational capacity and biological stability. *Mol Ther*, 16(11):1833–1840, 2008.

P Morais, H Adachi, and YT Yu. The critical contribution of pseudouridine to mrna covid-19 vaccines. front cell dev biol. *Front Cell Dev Biol*, 9:789427, 2021.

ONT-Dorado. Dorado. `https://github.com/nanoporetech/dorado`. Accessed: 2025-02-07.

ONT-Remora. Remora. `https://github.com/nanoporetech/remora`. Accessed: 2025-02-07.

A Radford, JW Kim, T Xu, G Brockman, C McLeavey, and I Sutskever. Robust speech recognition via large-scale weak supervision. *In International conference on machine learning*, pp. 28492–28518, 2023.

IA Roundtree, ME Evans, T Pan, and C He. Dynamic rna modifications in gene expression regulation. *Cell*, 1187-1200:169(7), 2017a.

IA Roundtree, GZ Luo, Z Zhang, X Wang, T Zhou, Y Cui, J Sha, X Huang, L Guerrero, P Xie, E He, B Shen, and C He. Ythdc1 mediates nuclear export of n-methyladenosine methylated mrnas. *eLife*, 6, 2017b. doi: 10.7554/eLife.31311.

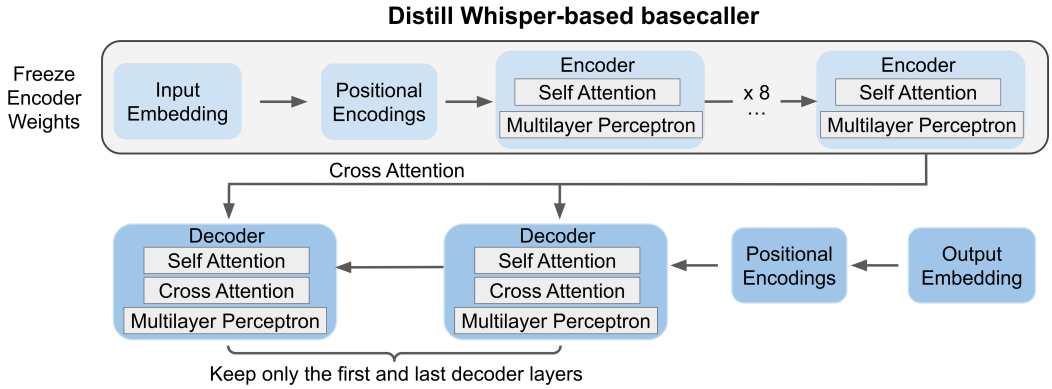
H Song, J Zhang, B Liu, J Xu, B Cai, H Yang, J Straube, X Yu, and T Ma. Biological roles of rna m5c modification and its implications in cancer immunotherapy. *Biomarker Research*, 10(1): 1–15, 2022.

YK Wan, C Hendra, PN Pratanwanich, and J Göke. Beyond sequencing: machine learning algorithms extract biology hidden in nanopore signal data. *Trends Genet*, 38(3):246–257, 2022.

Y Zhang, L Lu, and X Li. Detection technologies for rna modifications. *Experimental Molecular Medicine*, 54(10):1601–1616, 2022.

X Zhao, Y Zhang, D Hang, J Meng, and Z Wei. etecting rna modification using direct rna sequencing: A systematic review. *Comput Struct Biotechnol J.*, 20:5740–5749, 2022.

8 Supplementary Figure



Supplementary Figure 1. Distilling the arowana basecaller.