

# Training data diversity enhances the basecalling of novel RNA modification-induced nanopore sequencing readouts

Received: 16 May 2024

Accepted: 2 January 2025

Published online: 15 January 2025

Ziyuan Wang<sup>1,6</sup>, Ziyang Liu<sup>1,2,6</sup>, Yinshan Fang<sup>1,3,6</sup>, Hao Helen Zhang<sup>2,4</sup>, Xiaoxiao Sun<sup>2,5</sup>, Ning Hao<sup>1,2,4</sup>✉, Jianwen Que<sup>1,3</sup>✉ & Hongxu Ding<sup>1,2</sup>✉

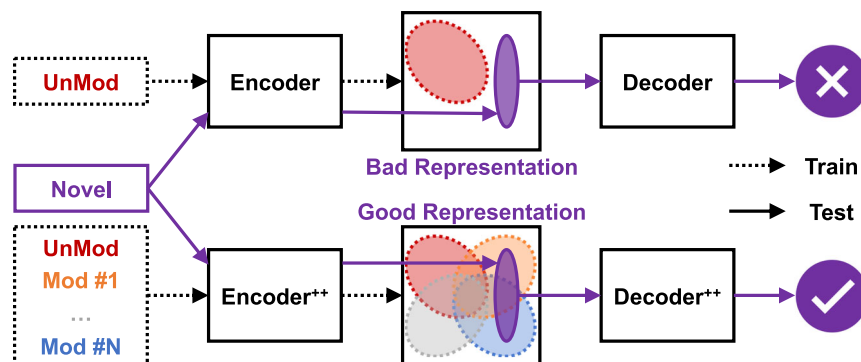
Accurately basecalling sequence backbones in the presence of nucleotide modifications remains a substantial challenge in nanopore sequencing bioinformatics. It has been extensively demonstrated that state-of-the-art basecallers are less compatible with modification-induced sequencing signals. A precise basecalling, on the other hand, serves as the prerequisite for virtually all the downstream analyses. Here, we report that basecallers exposed to diverse training modifications gain the generalizability to analyze novel modifications. With synthesized oligos as the model system, we precisely basecall various out-of-sample RNA modifications. From the representation learning perspective, we attribute this generalizability to basecaller representation space expanded by diverse training modifications. Taken together, we conclude increasing the training data diversity as a paradigm for building modification-tolerant nanopore sequencing basecallers.

During nanopore sequencing, biomolecules with various chemical structures translocate through protein pores further produce squiggling electric signals<sup>1</sup>. While such a rationale promises the routine detection of nucleotide chemical modifications, these modifications bring substantial challenges for accurately basecalling underlying sequence backbones. State-of-the-art basecallers, such as Guppy, Bonito, and Dorado delivered by the Oxford Nanopore Technologies, were built upon deep neural networks<sup>2</sup>. Extensive studies have reported that such basecallers are susceptible to modifications, which commonly exist in native DNA/RNA molecules. Most importantly, modification-induced basecalling artifacts are systematic, which cannot be resolved by simply increasing the sequencing depth as what we in general do for random errors. Indeed, these non-random errors can serve as informative signatures for determining well-studied modifications<sup>3–8</sup>. The large number of uncharacterized modifications further aggravate this problem: it is widely-believed that a majority of biologically-relevant nucleotide modifications remain undiscovered<sup>9,10</sup>; among the >50 DNA<sup>11</sup> and >170 RNA<sup>12</sup> modifications that have been discovered in vivo, most

of them have yet to be nanopore sequenced. Unlike the well-studied modifications, without prior biological knowledge, we are agnostic about the types and locations of basecalling errors produced by uncharacterized modifications. Taken together, these systematic and in most cases “cryptic” artifacts significantly undermine the basecalling accuracy.

The precise basecalling, on the other hand, serves as the prerequisite for virtually all the downstream bioinformatic analyses, including genome assembly and structural variation characterization, transcriptomic alternative splicing and expression quantification as well as DNA/RNA modification detection<sup>13</sup>. To better accomplish such biological applications, basecallers that are agnostic of modifications, especially the previously-uncharacterized ones, are therefore in pressing need. Here, we present a paradigm for basecalling previously-unseen modifications, by combining diverse existing modifications as training data. Specifically, the latest deep learning basecallers in general consist of encoder and decoder neural networks. The encoder network will condense sequencing signals into a highly-informative

<sup>1</sup>Department of Pharmacy Practice and Science, University of Arizona, Tucson, AZ, USA. <sup>2</sup>Statistics and Data Science GIDP, University of Arizona, Tucson, AZ, USA. <sup>3</sup>Columbia Center for Human Development, Department of Medicine, Columbia University Medical Center, New York, NY, USA. <sup>4</sup>Department of Mathematics, University of Arizona, Tucson, AZ, USA. <sup>5</sup>Department of Epidemiology and Biostatistics, University of Arizona, Tucson, AZ, USA. <sup>6</sup>These authors contributed equally: Ziyuan Wang, Ziyang Liu, Yinshan Fang. ✉e-mail: [nhao@math.arizona.edu](mailto:nhao@math.arizona.edu); [jq2240@cumc.columbia.edu](mailto:jq2240@cumc.columbia.edu); [hongxuding@arizona.edu](mailto:hongxuding@arizona.edu)



**Fig. 1 | Overview of the basecaller training strategy.** Basecallers in general consist of encoder and decoder neural networks: encoders first condense nanopore sequencing readouts as highly-informative representations; decoders further transform the produced representations as nucleotide sequences. Diverse training modifications will expand the representation space, thus making basecallers

generalizable to novel modifications. UnMod and Mod denote unmodified and modified training data categories, respectively. Novel denotes the out-of-sample modification in the test data. Encoder<sup>++</sup> and Decoder<sup>++</sup> comprise the basecaller trained with diverse modifications, as opposed to the basecaller trained with only the UnMod data.

representation space. Diverse training modifications can expand such a space, to a degree that out-of-sample modifications can also be properly encoded. By this means, previously-unseen modifications will be precisely basecalled by the decoder network, as shown in Fig. 1.

## Results

### An oligo-based model system for investigating nanopore sequencing basecalling

Generic basecallers, which in theory could handle any biological and artificial nucleotide sequences, are extremely compute-intensive and data-demanding to train. We therefore leverage control oligos as the model system to develop and evaluate basecallers. In line with previous studies<sup>3,6,14</sup>, our model system contains 4 oligo backbones, which together covered all 1024 RNA 5mers with a median occurrence of 10. These diverse sequence contexts were adopted in order to ensure the soundness of our basecalling analyses. To explore the basecalling of modification-induced nanopore sequencing readouts, besides unmodified (UM) oligos, 8 additional modified derivatives including N1-methyladenosine (m1A), N6-methyladenosine (m6A), N4-acetylcytidine (ac4C), 5-methylcytosine (m5C), 5-hydroxymethylcytosine (hm5C), 5-methyluridine (m5U), pseudouridine (Psi) as well as N1-methylpseudouridine (m1Psi), were collected for our analyses (see METHODS).

### Unmodified sequences are inadequate to train the modification basecaller

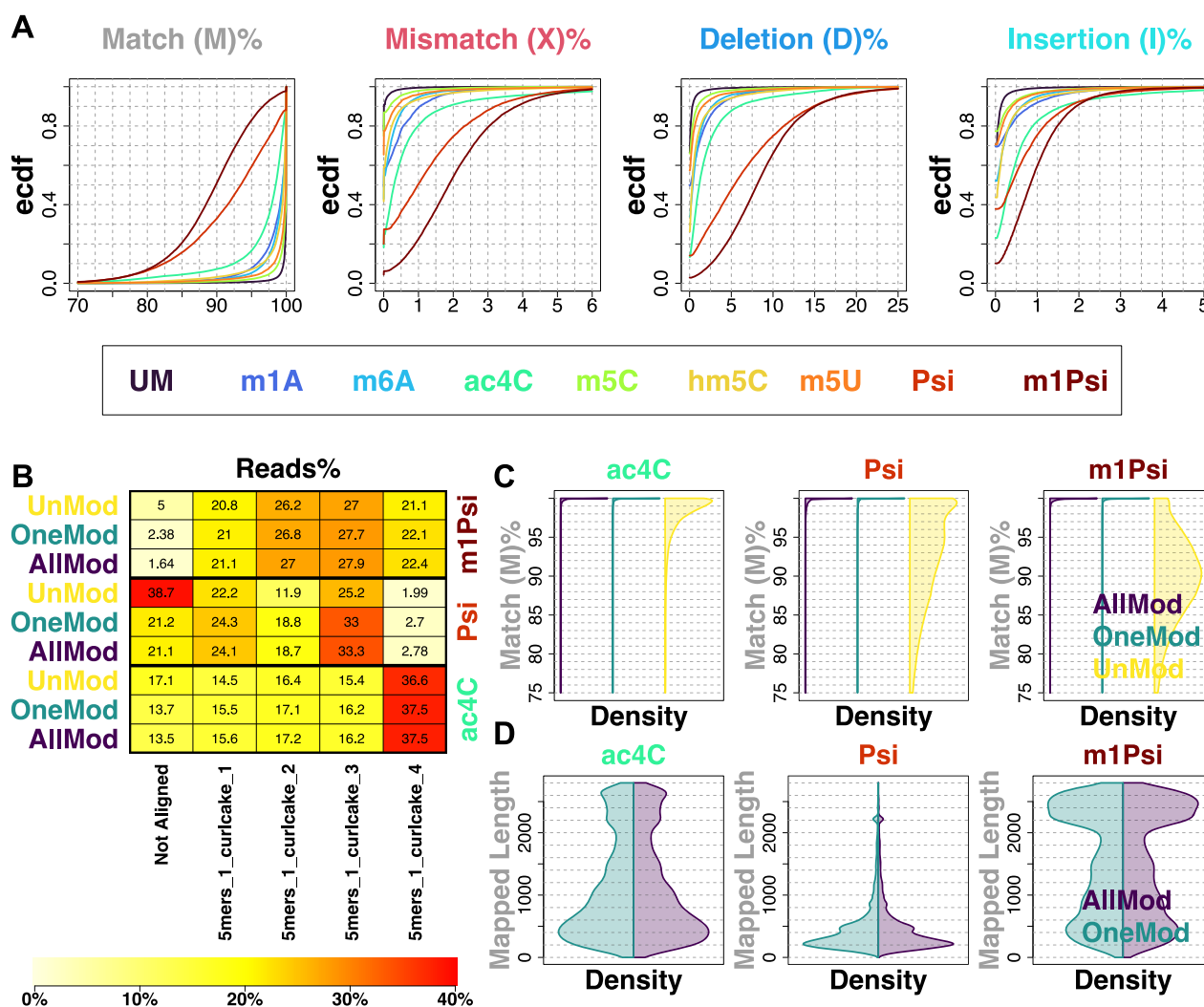
We first examined whether a basecaller trained using only unmodified nucleotide sequences can properly handle their modified counterparts. We therefore trained a basecaller using UM oligos, then executed it on UM, m1A, m6A, ac4C, m5C, hm5C, m5U, Psi, and m1Psi test oligos. We quantified the basecalling accuracy “functionally” with downstream alignment CIGAR (see METHODS). As the positive control, UM oligos were accurately basecalled with a 99.80% average match rate, which confirmed the high-quality basecaller training. We also noticed that m5C, hm5C, and m5U oligos were acceptably basecalled (99.48%, 98.57%, and 99.18% average match rate, respectively), which suggested the UM-trained basecaller could be generalized to a limited number of modifications. The remaining test groups, in particular ac4C, Psi, and m1Psi, drastically decreased basecalling confidence and produced considerably more basecalling errors (Fig. 2A). For instance, we found an average 2.23%, 6.45%, and 8.42% increase in deletion, which was the most common basecalling error in our analysis, for ac4C, Psi, and m1Psi compared to UM respectively. Taken together, such results demonstrated that basecallers trained with only unmodified sequences are less generalizable to analyze modification-induced nanopore readouts.

### Diverse training modifications improve the basecalling of novel modifications

We next asked whether combining diverse known modifications in the training dataset could facilitate the basecalling of novel modifications. We therefore investigated ac4C, Psi, and m1Psi, which were identified as “least analyzable” modifications with the UM-basecaller. Specifically, we trained corresponding basecallers with all other modifications except for the ones to be analyzed (the “AllMod” groups). Meanwhile, we trained basecallers using only candidate modifications (the “OneMod” groups) as positive controls. We also used the above UM-basecaller (the “UnMod” groups) to evaluate the baseline basecalling performance (see METHODS). As shown in Fig. 2B, “AllMod” generated comparable mappability with “OneMod” positive controls for all the three modification types, with absolute differences <0.8%. As shown in Fig. 2C, “AllMod” increased basecalling accuracy compared to “UnMod”: we noticed an average increase of 2.60%, 7.30%, and 10.40% in CIGAR matches for ac4C, Psi, and m1Psi, respectively. Most importantly, “AllMod” basecalling accuracy reached the same level as “OneMod” positive controls: we noticed a negligible <0.40% absolute difference for all the three oligo types. Finally, as shown in Fig. 2D, the “AllMod” training will polish, rather than biasing basecalling, by generating consistent mapped lengths compared to the “OneMod” ground-truth. In summary, our results demonstrated that increasing training modification diversity will enhance the basecalling performance of out-of-sample novel modifications. We further confirmed such a conclusion with the latest Bonito and Dorado basecalling systems, as shown in Fig. S1.

### Evaluating the out-of-sample basecalling generalizability for training modification combinations

We further asked, are all the training modifications required for accurate out-of-sample basecalling? To formally answer such a question, an evaluation metric for training modification combinations is required. Inspired by our discovery that basecallers trained with only unmodified oligos basecalled certain modifications (m5C, hm5C, m5U) with acceptable accuracy (Fig. 2A), as well as signals of such modifications were less deviated from their canonical counterpart (Fig. S2, see METHODS), we hypothesized that the optimal training combinations can completely cover signals of the out-of-sample modification. We therefore defined the “signal cover score”, with the following five steps: (1) correspond signal segments (events) with basecalled sequences; (2) assemble all the events mapped to the same sequence position, calculate their signal mean values, then use 10% and 90% quantiles (to avoid outliers) as the “effective signal range”; (3) quantify the training effective signal range, by taking the union of all the training modifications; (4) measure the cover



**Fig. 2 | Basecallers trained with diverse known modifications gain the capability to basecall novel modifications.** **A** Performance of the basecaller trained only by the unmodified data on all the read groups. Basecalling performance was assessed with the per-read CIGAR alignment fraction, including match (M), mismatch (X), deletion (D) and insertion (I). UM and acronyms stand for unmodified and modified RNA oligo categories, respectively. Ecdf denotes the empirical cumulative distribution function. Performance of basecallers trained by combining

all the oligo groups except for ac4C, Psi or m1Psi was quantified. Specifically, the mappability (**B**) and per-read CIGAR match fraction (**C**) were used as quantification metrics. AllMod, the basecaller trained by all the modifications except for the one to be basecalled; OneMod, the basecaller trained with only the modification to be basecalled; UnMod, the basecaller trained by only unmodified reads. **D** Mapped length distributions of “AllMod” and “OneMod”.

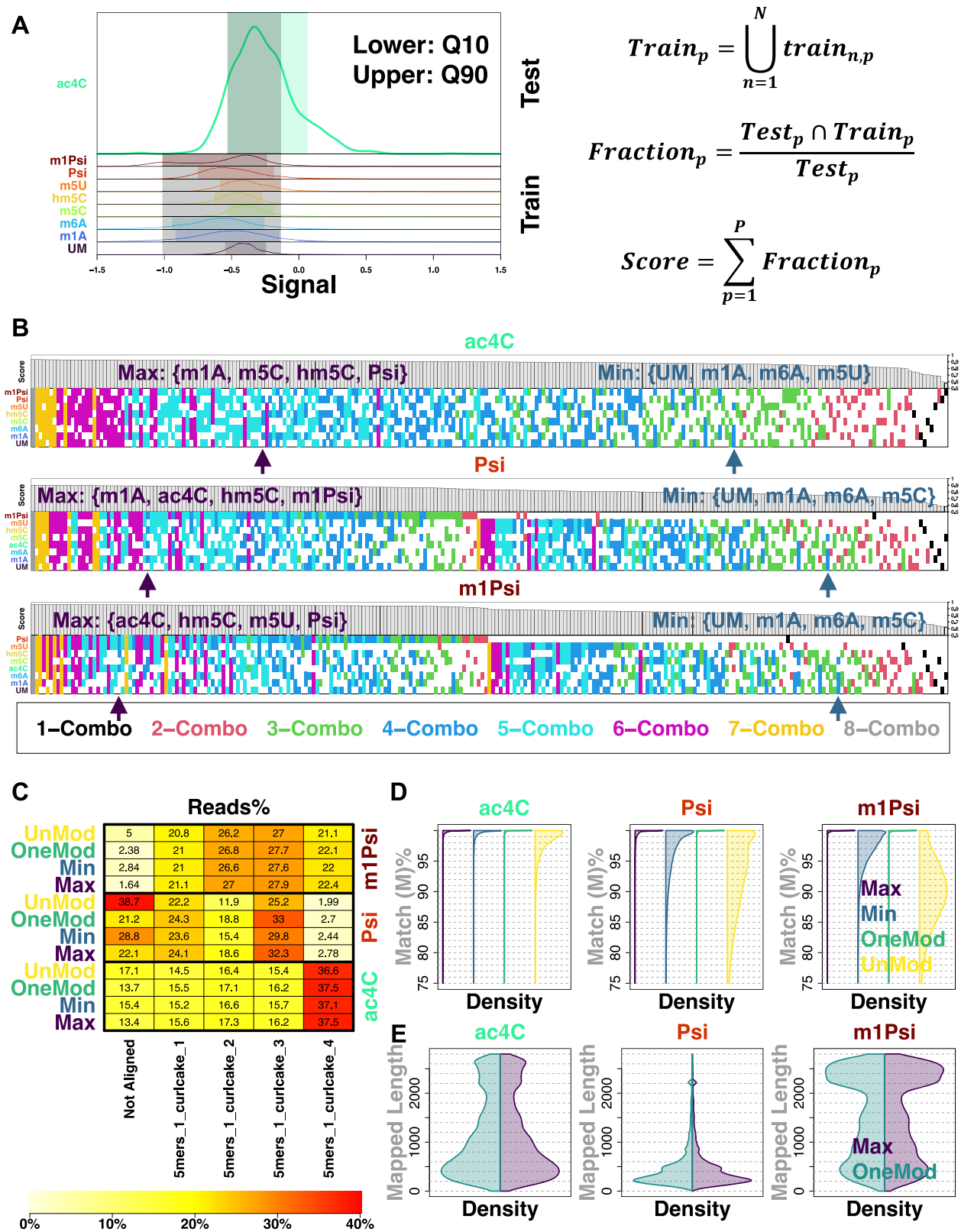
fraction, as the test effective signal range that can be covered by the training counterpart; (5) calculate the final signal cover score, by taking the summation of all the cover fractions along the candidate sequence (Fig. 3A).

Based on such a signal cover score, for ac4C, Psi, and m1Psi test groups, we evaluated all the possible training modification combinations. As shown in Fig. 3B, we observed that combinations of more modifications in general produced higher signal cover scores, which echoes with our observation that diverse training data improves the basecalling of out-of-sample modifications. In particular, we noticed that the inclusion of Psi and m1Psi in training modifications significantly improved signal cover scores of m1Psi and Psi test data, respectively. These findings echo with our results that Psi and m1Psi reads can be acceptably handled by basecallers training using only m1Psi and Psi reads, respectively (Figs. S4A and S5A). We further systematically confirmed that signal cover scores can be adopted to evaluate training modification combinations. Without losing generality, we prioritized the analyses of 4-combo combinations, and highlighted ones with the highest (Max) and lowest (Min) scores. We found that “Max” generated

comparable mappability and basecalling accuracy with “OneMod” positive controls (basecallers trained with only candidate modifications), and out-performed “Min”, particularly in Psi and m1Psi groups, as shown in Fig. 3C, D. We finally confirmed that “Max” could generate consistent mapped lengths with “OneMod” ground-truth, as shown in Fig. 3E. Therefore, we concluded the signal cover score as the metric for prioritizing training modification combinations.

### Precise basecalling requires high-quality data representations

We then related the basecaller accuracy with the quality of its encoder representation space. Representation learning condenses neural network inputs into a highly-informative representation space to achieve downstream tasks<sup>15</sup>. During basecalling, nanopore sequencing signals will be encoded in the representation space then decoded as nucleotide sequences. To explore how data representations affect the basecalling accuracy, we analyzed ac4C test oligos. In particular, we trained the “All” basecaller by combining all the oligo categories (except for ac4C), as well as 8 single-category (without ac4C) basecallers (see METHODS).



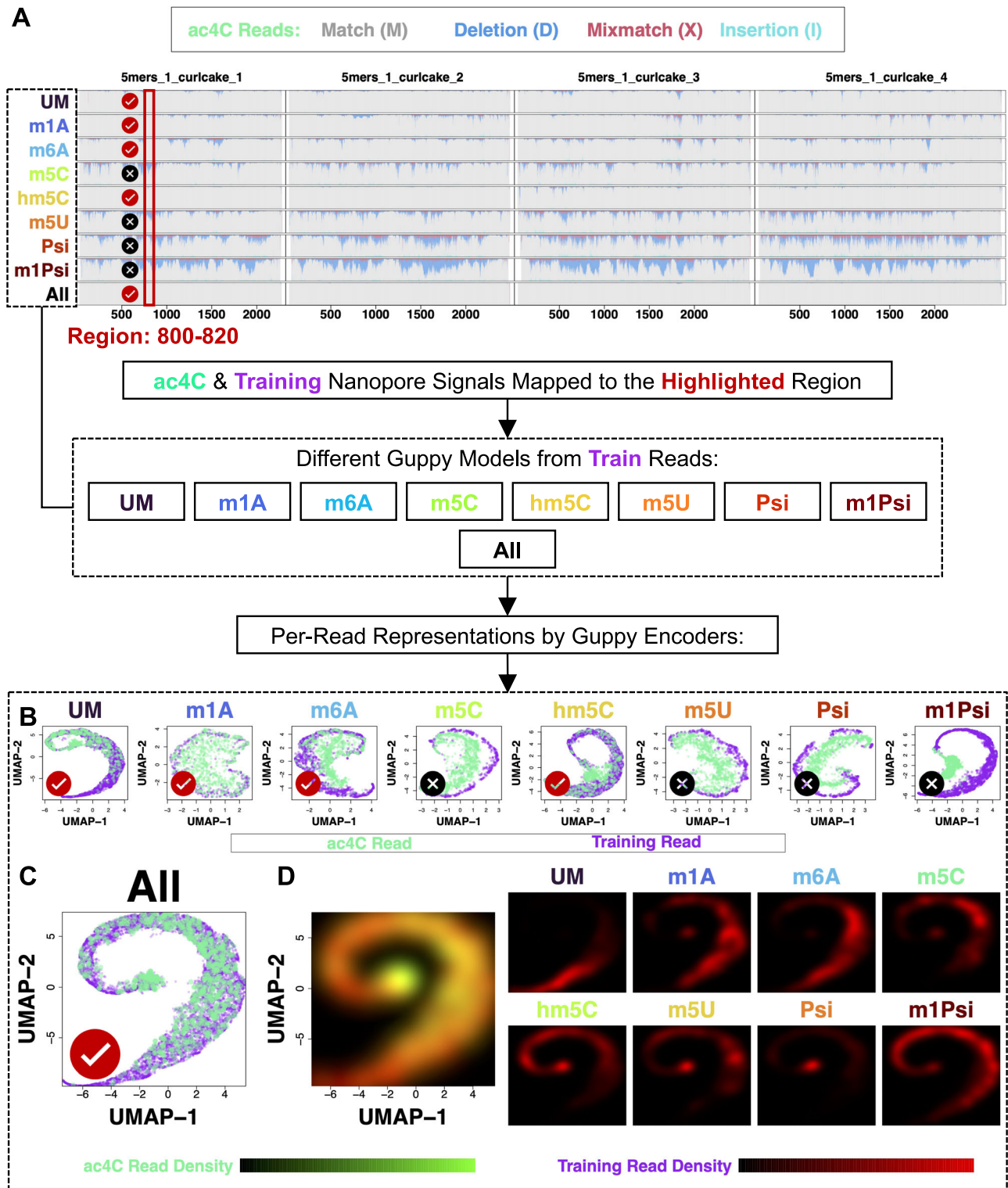
**Fig. 3 | Prioritizing training combinations for precise out-of-sample modification basecalling using signal cover scores.** **A** The definition of “signal cover score”. Q10 and Q90 mark the 10% and 90% signal quantiles, respectively.  $N$  and  $P$  denote the total number of training modification classes and sequence positions, respectively. **B** Signal cover scores for all the possible training modification combinations in descending order, for ac4C, Psi, and m1Psi test groups. Bars denote the inclusion of training modifications, and corresponding colors represent numbers of modifications, for a certain

combination. Specifically, 4-combo combinations with the highest (Max) and lowest (Min) signal cover scores were marked. The performance of Max and Min basecallers was next quantified. Specifically, the mappability (**C**) and per-read CIGAR match fraction (**D**) were used as quantification metrics. AllMod, the basecaller trained by all the modifications except for the one to be basecalled; UnMod, the basecaller trained by only unmodified reads. **E** Mapped length distributions of “AllMod” and “Max”.



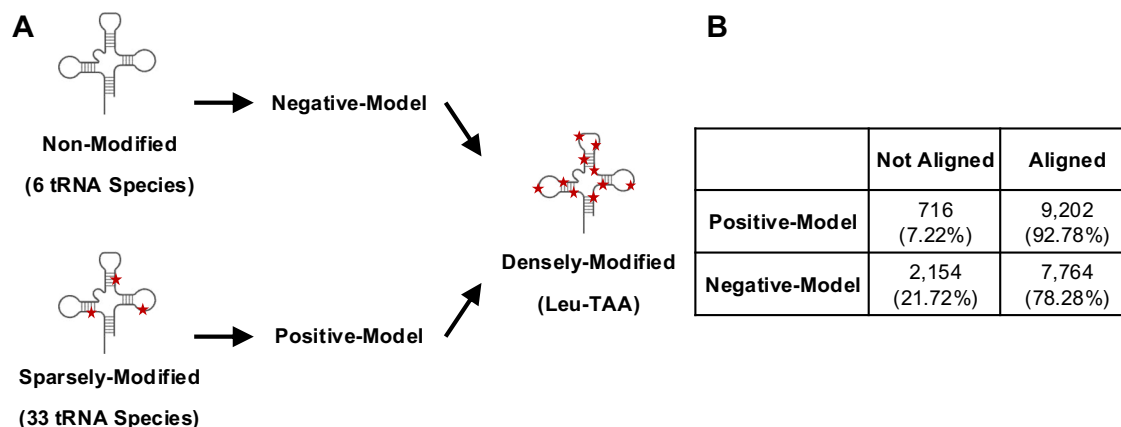
We observed that compared to the individually-trained basecallers, “All” can significantly promote the CIGAR match fraction. Such precise basecalling was consistently observed in different regions among all the four oligos. We further observed that although artifacts

made by individually-trained basecallers were in general prevalent, certain regions were more likely to be accurately analyzed. For example, “m1A” decently analyzed the region 800 to 820 of the first oligo, which was highlighted with the red box (Fig. 4A). We next



**Fig. 4 | Diverse training data expands the representation space thus making the basecaller generalizable to novel modifications.** **A** Performance of individually and jointly-trained basecallers on ac4C reads was visualized with the genome viewer graph, which shows per-nucleotide CIGAR fractions. All, the jointly-trained basecaller by all the oligo types except for ac4C; other acronyms denote individually-trained basecallers. For individually (**B**) and jointly-trained (**C**)

basecallers, read fragments mapped to the boxed region were first converted as representation vectors with different basecaller encoders, then visualized by a UMAP plot. Train denotes reads used for training the corresponding basecaller. **D** Spatial distributions of different oligo types in the UMAP space as shown in (**C**). Black-to-green and red palette denotes ac4C and training reads, respectively.



**Fig. 5 | Diverse sparsely-modified training tRNAs enable the precise basecalling of the densely-modified Leu-TAA.** **A** Overview of the analysis. **B** The mappability of Leu-TAA tRNA reads. A total of 9918 Leu-TAA reads were analyzed.

investigated such region-specific elevated basecalling performance, by interrogating the encoder representation space. Without losing generality, we prioritized the above boxed region, by projecting corresponding nanopore signals into the representation space (see METHODS). Within the representation space, we quantified the read-level similarity and found that, in the more accurate “UM”, “m1A”, “m6A”, and “hm5C” basecallers, ac4C test reads were more similar to their corresponding training reads. On the contrary, test-train similarity was significantly reduced among error-prone “m5C”, “m5U”, “Psi”, and “m1Psi” basecallers. Most importantly, the highest test-train similarity was produced by the most precise “All” basecaller (Fig. S3A). We further visualized such a representation space similarity pattern between test and training reads with UMAP, as shown in Fig. 4B. In representation learning, test data points can be properly encoded, if and only if they fall in the manifold produced by training data points. We therefore highlight the importance of a high-quality representation space in precise basecalling. We further confirmed this conclusion with Psi (Figs. S3B and S4A–C) and m1Psi (Figs. S3C and S5A–C) test oligos.

### Training data diversity yields a generalizable basecaller representation space

We further explained, from a representation learning perspective, that diverse training oligos will expand the representation space, to a degree that out-of-sample novel modifications could also be properly encoded. Specifically, we revisited the “All” representation space, and assessed the spatial distribution of ac4C, as well as different types of training oligos (Fig. 4D). We presented the spatial density of ac4C and training data points using the black-to-green and black-to-red palette, respectively. We first confirmed that the majority of ac4C points were covered with the training manifold, by finding negligible stand-alone green area. We further found that the entire training manifold was required to thoroughly encode ac4C data, by finding most representation space to be yellowish. We also found that training groups occupied different sub-space, which together completed the training manifold. Taken together, these results suggested that diverse training modifications will complement each other, thereby generalizing the representation space to out-of-sample modifications. We further confirmed such a conclusion with Psi (Fig. S4D) and m1Psi (Fig. S5D) test oligos.

### Training data diversity enhances the basecalling of densely-modified tRNA reads

Finally, we demonstrated the practical usefulness of our basecaller training paradigm by analyzing a yeast native tRNA nanopore sequencing dataset<sup>16</sup>. Specifically, we aimed to precisely basecall the most densely-modified Leu-TAA tRNAs, which contains 15 known

modification sites. We further trained positive and negative-models, by combining the 33 sparsely-modified and 6 non-modified tRNA species, respectively (see METHODS). Our workflow was summarized in Fig. 5A. We quantified the basecalling performance with mappability. We found that, compared to the negative-model which could only represent canonical sequences, the positive-model trained using diverse modifications achieved a ~15% increase in mappability (Fig. 5B). Therefore, we highlighted our paradigm to be a general way of training modification-tolerant basecallers.

## Discussion

Promoting the basecalling accuracy, especially for native DNA/RNA sequencing signals, remains a central challenge in nanopore sequencing bioinformatics. This is because the latest basecallers are less tolerant to modifications, which commonly exist among native DNA and RNA molecules and will deviate nanopore sequencing signals. Specifically, we observed that shifted signals primarily cause mis-basecalls, while changes in dwell time mainly affect basecalled sequence lengths. For instance, for Psi and m1Psi, significantly deviated signals (quantified by mean and standard deviation) and dwell time (Fig. S2) coincided with excessive mismatches and deletions/insertions, respectively (Fig. 2A). As for ac4C, the presence of which particularly alters dwell time (Fig. S2), we noticed insertions as the major type of basecalling artifacts (Fig. 2A). To properly address this limitation, basecallers that are agnostic to nucleotide modifications are in urgent need.

Here, we presented a paradigm for training such modification-tolerant basecallers: we demonstrated that previously-unseen modifications could be precisely basecalled by including diverse existing modifications in training data. We anticipated such a paradigm increasing the basecalling accuracy of native DNA/RNA nanopore sequencing readouts, further shedding light on diverse biological usecases, e.g., de novo genome assembly by generating highly-accurate DNA contigs<sup>17</sup>, mRNA vaccine quality analyses by rigorously assessing sequence, length, integrity and purity<sup>18</sup>, and DNA/RNA modification detection by delivering precise backbones to scrutinize modification status of each nucleotides<sup>19</sup>.

To fully leverage such a paradigm, we emphasized the quality and generalizability of the basecaller representation space. In particular, a high-quality representation space could extract all essential information for a precise basecalling; a generalizable representation space could tolerate sequencing signal variations for the novel modification basecalling. To optimize the basecaller representation space, as the potential future direction, we will leverage self-supervised learning techniques<sup>20</sup> for training foundation model encoders<sup>21</sup>. During the self-supervised learning process, the encoder neural network first transforms inputs

into a representation space. The decoder neural network then reconstructs inputs from the encoded representation space. The recreation of inputs suggests the negligible information loss in the representation space. By this means, all the information harbored inside nanopore sequencing signals could be extracted by the encoder, further providing a solid foundation for the precise basecalling. Foundation models are trained by a broad range and excessive amount of data, therefore can be applied across diverse usecases. This paradigm echoes with our discovery, that diverse training oligos will expand, further generalize the representation space for basecalling various out-of-sample modifications. Altogether, we expect the foundation model encoder trained via self-supervised learning to be able to produce the high-quality and generalizable representation space, therefore enhancing the performance of modification-tolerant basecalling.

## Methods

### RNA oligo synthesis

RNA oligo sequences (“curlcakes”) were reported in ref. 3, and were cloned into engineered pUC57 in vitro transcription (IVT) plasmids. We then incorporated modified nucleotides into curlcakes with IVT. We produced unmodified and m6A RNA oligos for resequencing because existing datasets<sup>3</sup> were sequenced over 5 years ago and may be outdated. We also included m1A, which was not surveyed by previous studies<sup>3,6,14</sup>, in our modification collection. Specifically, curlcake-containing pUC57 plasmids were digested using EcoRV (New England Biolabs, R0195L) and BamHI (New England Biolabs, R0136L) restriction enzymes for at least two hours at 37 °C, and further analyzed with agarose gel electrophoresis. The digested DNA was purified by the PCR purification kit (QIAGEN, 28104), as the template for IVT. NanoDrop Spectrophotometers (Thermo Fisher Scientific) was used to measure the concentration of extracted linear DNA prior to IVT. Ampliscribe™ T7-Flash™ Transcription Kit (BioSearch Technologies, ASF3507) was used to generate IVT RNAs as per manufacturer’s instructions. The four canonical (ATP, CTP, GTP, and UTP) included in the Ampliscribe™ T7-Flash™ Transcription Kit) ribonucleoside triphosphates were supplemented during IVT for producing unmodified RNA oligos. Meanwhile, modified ribonucleoside triphosphates including N1-Methyl-ATP (m1A, TriLink Biotechnologies, N-1042) and N6-Methyl-ATP (m6A, TriLink Biotechnologies, N-1013) were supplemented in place of their unmodified counterparts for producing modified RNA oligos. DNase I (included in the Ampliscribe™ T7-Flash™ Transcription Kit) was added to the IVT reaction system after incubation for 4 h at 42 °C to eliminate the residual template DNA. Yielded IVT RNAs were purified using the RNeasy Mini Kit (QIAGEN, 74104) following manufacturer’s instructions. Vaccinia capping enzyme (New England Biolabs, M2080S) was used for the 5’ capping of purified IVT RNAs, with an incubation for 30 min at 37 °C. Following purification with RNAClean XP Beads (Beckman Coulter, A63987), the capped IVT RNAs were subjected to polyadenylation tailing (New England Biolabs, M0276L). Concentration of capped and polyA-tailed IVT RNAs was determined by Qubit Fluorometer (Thermo Fisher Scientific).

### Nanopore sequencing

RNA nanopore sequencing libraries were built using the ONT Direct RNA Sequencing Kit (Oxford Nanopore Technologies, SQK-RNA002) following protocol version DRS\_9080\_v2\_revQ\_14Aug2019 as per manufacturer’s instructions. Briefly, for each RNA curlcake category, 2 µg of capped and polyA-tailed IVT RNA was subjected to adapter ligation using the T4 DNA Ligase (New England Biolabs, M0202M), followed by reverse transcription using the SuperScript III Reverse Transcriptase (Thermo Fisher Scientific, 18080044). After purification using RNAClean XP Beads (Beckman Coulter, A63987), yielded RNA:DNA hybrids were ligated to RNA adapters using the T4 DNA Ligase (New England Biolabs, M0202M). The concentration of the yielded RNA library was determined using the Qubit Fluorometer

(Thermo Fisher Scientific). The RNA library was mixed with the RNA Running Buffer prior to sequencing on a primed Flongle flowcell. The flowcell version is R9.4.1, and the sequencer is MinION with a Flongle adapter.

### Creating ground-truth sequence labels for RNA oligos

We used iterative basecalling to generate ground-truth sequence labels for RNA oligos. During iterative basecalling, we randomly sampled ~20,000 reads for each modification type (unmodified, m1A, m6A, ac4C, m5C, hm5C, m5U, Psi, m1Psi) and combined them together as a single training dataset. We used Guppy (version 6.0.6 + 8a98bbc), Taiyaki (version 5.3.0) and Samtools (version 1.16) to perform iterative basecalling. Specifically, we used the Guppy basecalling configuration “rna\_r9.4.1\_70bps\_hac.cfg” for the initial iteration, and the model trained from the previous iteration for the subsequent iteration. The “--disable\_qscore\_filtering” flag was set to keep “low-quality reads” that are usually artifacts caused by modifications. Samtools functions merge, sort and index with default flags were used to process alignment results generated by Guppy. Taiyaki was used to train basecalling models that are compatible with Guppy. For training data preparation, get\_refs\_from\_sam.py with flag “--reverse”, generate\_per\_read\_params.py with default flags, prepare\_mapped\_reads.py with the Taiyaki “mLstm\_flipflop\_model\_r941\_DNA” checkpoint file, and merge\_mappedsignalfiles.py with default flags were used. For training Guppy models, train\_flipflop.py with flags “--size 256 --stride 10 --winlen 31” and the model template “mLstm\_cat\_mod\_flipflop.py” were used. For preparing Guppy models, dump\_json.py with default flags on the final model checkpoint were used. We performed a total of 4 iterations to guarantee the labeling accuracy, and the comparison between original Guppy and iteratively-optimized basecallers was shown in Fig. S6.

### Guppy basecaller training and basecalling analysis for curlcakes

We trained a total of 18 Guppy basecalling models, including 9 with single modification categories (unmodified, m1A, m6A, ac4C, m5C, hm5C, m5U, Psi, m1Psi), 3 “leave-out” models (trained with all modifications except for ac4C, Psi or m1Psi), as well as 3 “Max” and 3 “Min” models (described in Fig. 3B). Training data for individual modification types were prepared from the last iteration of the above-described iterative basecalling process. For “leave-out” models, individual training datasets were combined using the Taiyaki merge\_mappedsignalfiles.py function, as above-described. Basecaller models were next trained using the Taiyaki train\_flipflop.py and dump\_json.py functions, as described in the above section.

Guppy basecalling was performed on independent test datasets: we randomly sampled ~20,000 reads for each modification type, same as the construction of training datasets. We used flags specified in the configuration file “rna\_r9.4.1\_70bps\_hac.cfg”, except for providing our own models for the Guppy basecalling analysis. Basecalling accuracy was “functionally” evaluated using per-read alignment results, including mappability (the ratio between aligned reads regardless of their SAM flags, as opposed to all the reads), and fractions of CIGAR M (match), X (mismatch), D (deletion) and I (insertion). Specifically, for “leave-out” analyses, we also examined alignment consistency (alignment start and end positions) as opposed to “self” models (model trained with the left-out modification). Alignment analyses were performed with the built-in minimap2 aligner of Guppy, with all the alignment setups set as default.

### Bonito and dorado basecaller training and basecalling analysis for curlcakes

We trained a total of 9 Bonito basecalling models, including 9 with single modification categories (unmodified, m1A, m6A, ac4C, m5C, hm5C, m5U, Psi, m1Psi), 3 “leave-out” models (trained with all modifications except for ac4C, Psi or m1Psi). Training data for individual modification types was prepared from the last iteration of the above-



described iterative basecalling process. Basecaller models were next trained with the “bonito train” command using the same model architecture as the “rna002\_70bps\_hac@v3” model. We used the “bonito basecaller” command to basecall test datasets. We used the “bonito export” command to transform the Bonito model to Dorado model and used the “dorado basecaller” command to basecall the same test datasets as in the Guppy analyses were used. Bonito version 0.8.1 and Dorado version 0.7.0+71cc744 were used in these analyses.

### Representation space analysis

Representation space analyses were performed on all possible pairs of basecallers, e.g., above-mentioned single and “leave-out” models and test oligo types, e.g., ac4C, Psi and m1Psi. For a specific pair, e.g., the ac4C “leave-out” model and ac4C test data as shown in Fig. 2, we first extracted test sequencing signal chunks that could be approximately corresponded to region curlcake\_1.5mers.1:800–820 by the basecaller. Specifically, we scanned consecutive signal chunks of 2000 data points, and collected ones whose first basecalled nucleotides aligned inside curlcake\_1.5mers.1:795–800. We subsequently flowed such chunks through the basecaller encoder neural network, in order to produce a latent feature-by-single chunk matrix in the representation space. With such a matrix, we then performed Principal Component Analysis, and took the first 50 PCs for the final Uniform Manifold Approximation and Projection (UMAP) visualization<sup>22</sup>.

### Nanopore sequencing signal analysis

We first basecalled nanopore sequencing reads using Bonito (version 0.8.1) to produce move-tables, which track signal chunks corresponding to individual nucleotides (stored as mv tags in the generated bam files). We next executed the Remora (version 3.2.0) pipeline “Reference Region Metric Extraction” ([https://github.com/nanoporetech/remora/blob/master/notebooks/metrics\\_api.ipynb](https://github.com/nanoporetech/remora/blob/master/notebooks/metrics_api.ipynb)) to extract the per-nucleotide signal features (mean, standard deviation and dwell time). Throughout the analysis, all the Bonito and Remora parameters were set as default.

### Yeast tRNA analysis

Yeast tRNA reads were first iteratively basecalled and aligned as described in ref. 19. Reads were next grouped by alignment results and further used for model training. The positive and negative-models were trained by the Guppy workflow. Detailed training procedures and parameters were the same as those described in ref. 19.

### Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

### Data availability

The ac4C, m5U, and m1Psi RNA oligo nanopore sequencing datasets were downloaded from ENA under the accession number [PRJEB67632](#). The m5C RNA oligo nanopore sequencing dataset was downloaded from NCBI under the BioProject [PRJNA563591](#). The hm5C RNA oligo nanopore sequencing dataset was downloaded from NCBI under the BioProject [PRJNA548268](#). The Psi RNA oligo nanopore sequencing dataset was downloaded from NCBI under the BioProject [PRJNA549001](#). Resequenced unmodified, and m6A, as well as newly generated m1A nanopore sequencing datasets are available at NCBI under the BioProject [PRJNA1050579](#). Sequence backbones for RNA oligos were provided in the Supplementary Note 1 of ref. 3. The yeast native tRNA nanopore sequencing data was downloaded from ENA under accession number [PRJEB55684](#). Corresponding reference sequences and modification annotations were downloaded from <https://github.com/novoalab/Nano-tRNAseq/tree/main/ref>. Source data is provided at <https://doi.org/10.25422/azu.data.27976647>.

### Code availability

The workflow is publicly available and has been deposited in GitHub at <https://github.com/wangziyuan66/NanoRL>, under MIT License. The specific version of the code associated with this publication is archived in Zenodo and is accessible via <https://doi.org/10.5281/zenodo.14278318><sup>23</sup>. Users are permitted to reuse, modify, and distribute the code in accordance with the terms of the license. Any modifications to the code should appropriately credit the original authors as outlined by the license terms.

### References

- Deamer, D., Akeson, M. & Branton, D. Three decades of nanopore sequencing. *Nat. Biotechnol.* **34**, 518–524 (2016).
- Pagès-Gallego, M. & de Ridder, J. Comprehensive benchmark and architectural analysis of deep learning models for nanopore sequencing basecalling. *Genome Biol.* **24**, 71 (2023).
- Liu, H. et al. Accurate detection of m6A RNA modifications in native RNA sequences. *Nat. Commun.* **10**, 4079 (2019).
- Parker, M. T. et al. Nanopore direct RNA sequencing maps the complexity of Arabidopsis mRNA processing and m6A modification. *Elife* **9**, e49658 (2020).
- Price, A. M. et al. Direct RNA sequencing reveals m6A modifications on adenovirus RNA are necessary for efficient splicing. *Nat. Commun.* **11**, 6016 (2020).
- Begik, O. et al. Quantitative profiling of pseudouridylation dynamics in native RNAs with nanopore sequencing. *Nat. Biotechnol.* **39**, 1278–1291 (2021).
- Jenjaroenpun, P. et al. Decoding the epitranscriptional landscape from native RNA sequences. *Nucleic Acids Res.* **49**, e7 (2021).
- Nguyen, T. A. et al. Direct identification of A-to-I editing sites with nanopore native RNA sequencing. *Nat. Methods* **19**, 833–844 (2022).
- Raiber, E. et al. Mapping and elucidating the function of modified bases in DNA. *Nat. Rev. Chem.* **1**, 0069 (2017).
- Schaefer, M., Kapoor, U. & Jantsch, M. F. Understanding RNA modifications: the promises and technological bottlenecks of the ‘epitranscriptome’. *Open Biol.* **7**, 170077 (2017).
- Sood, A. J., Viner, C. & Hoffman, M. M. DNAmdb: the DNA modification database. *J. Cheminform.* **11**, 1–10 (2019).
- Boccalletto, P. et al. MODOMICS: a database of RNA modification pathways. 2021 update. *Nucleic Acids Res.* **50**, D231–D235 (2022).
- Wang, Y. et al. Nanopore sequencing technology, bioinformatics and applications. *Nat. Biotechnol.* **39**, 1348–1365 (2021).
- Diensthuber, G. et al. Enhanced detection of RNA modifications and read mapping with high-accuracy nanopore RNA basecalling models. *Genome Res.* **34**, 1865–1877 (2024).
- Bengio, Y., Courville, A. & Vincent, P. Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 1798–1828 (2013).
- Lucas, M. C. et al. Quantitative analysis of tRNA abundance and modifications by nanopore RNA sequencing. *Nat. Biotechnol.* **42**, 72–86 (2024).
- Sereika, M. et al. Oxford Nanopore R10. 4 long-read sequencing enables the generation of near-finished bacterial genomes from pure cultures and metagenomes without short-read or reference polishing. *Nat. Methods* **19**, 823–826 (2022).
- Gunter, H. M. et al. mRNA vaccine quality analysis using RNA sequencing. *Nat. Commun.* **14**, 5663 (2023).
- Wang, Z. et al. Adapting nanopore sequencing basecalling models for modification detection via incremental learning and anomaly detection. *Nat. Commun.* **15**, 7148 (2024).
- Liu, X. et al. Self-supervised learning: generative or contrastive. *IEEE Trans. Knowl. Data Eng.* **35**, 857–876 (2021).
- Bommasani, R. et al. On the opportunities and risks of foundation models. Preprint at *arXiv arXiv:2108.07258* (2021).



22. McInnes, L., Healy J. & Melville J. Umap: uniform manifold approximation and projection for dimension reduction. Preprint at *arXiv* arXiv:1802.03426 (2018).
23. Wang Z. Training data diversity enhances the basecalling of novel RNA modification-induced nanopore sequencing readouts. <https://github.com/wangziyuan66/NanoRL>. <https://doi.org/10.5281/zenodo.14278318> (2024).

## Acknowledgements

We thank the University of Arizona High Performance Computing team and the College of Pharmacy Information Technology Group for their support. H.D. is supported by HL166330, the University of Arizona Health Sciences Career Development Award and the University of Arizona Accelerate For Success Award. J.Q. is supported by HL159675, HL152293, AI163753, and DK132251.

## Author contributions

H.D. conceived the idea. Z.W., Z.L. and H.D. performed the analysis. Y.F. performed the experiment. H.H.Z., X.S., N.H., J.Q. and H.D. supervised the project. Z.W., Y.F. and H.D. wrote the manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s41467-025-55974-z>.

**Correspondence** and requests for materials should be addressed to Ning Hao, Jianwen Que or Hongxu Ding.

**Peer review information** *Nature Communications* thanks the anonymous reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

**Reprints and permissions information** is available at <http://www.nature.com/reprints>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025