OXFORD

Briefings in Bioinformatics, 2026, **27(1)**, bbaf709
https://doi.org/10.1093/bib/bbaf709
Case Study

Ab initio detection of multiple epitranscriptomic modifications from Oxford nanopore technology direct RNA sequencing data

Adriano Fonzino[1], Bruno Fosso[1], Grazia Visci[1], Carmela Gissi[1,2], Graziano Pesole[1,2], Ernesto Picardi[1,2,*]

[1]Department of Biosciences, Biotechnology, and Environment, University of Bari Aldo Moro, Via Orabona 4, 70125, Bari, Italy
[2]Institute of Biomembranes, Bioenergetics and Molecular Biotechnology, National Research Council, Via Amendola 122/o, 70126, Bari, Italy

*Corresponding author. Department of Biosciences, Biotechnology, and Environment, University of Bari Aldo Moro, Via Orabona 4, 70125, Bari, Italy. Tel: +390805442179. E-mail: ernesto.picardi@uniba.it

Abstract

Charting the eukaryotic epitranscriptome by direct RNA sequencing is promising but still very challenging, as current bioinformatics tools are based on modification-unaware software and require multiple modification-specific learning steps. Here, we introduce NanoSpeech, a modification-aware basecaller for the *ab initio* simultaneous detection of multiple modified bases using a transformer model, and NanoListener, which implements a simulated randomers strategy for robust training datasets and a new generation of ONT basecallers. NanoListener and NanoSpeech are independent of the specific ONT chemistry. Once a training dataset has been created, a single model with an expanded vocabulary can accurately basecall both unmodified and modified bases.

Keywords: RNA modifications; direct RNA sequencing; RNA editing; epitranscriptomics; ONT RNA basecalling

Introduction

With more than 170 different RNA chemical modifications, epitranscriptomics amazingly enhances the complexity of the eukaryotic transcriptomes [1], contributing to fine-tuning gene expression and regulation [2, 3]. Nowadays, complete epitranscriptome maps are yet elusive or in draft mode, and charting modified ribonucleosides remains challenging. Direct RNA sequencing (dRNA) through Oxford Nanopore Technology (ONT) [4, 5] holds the promise to unveil the vast repertoire of RNA modifications. According to ONT, individual RNA molecules translocate through engineered protein pores, producing squiggling electric signals depending on their specific ribonucleotide sequences [4, 5]. Next, base-calling algorithms, implementing complex deep-learning models, convert raw current signals into nucleotide sequences for downstream analyses. Most of the available tools for detecting RNA modifications from dRNA data [6–8] take into account systematic base miscalls due to modification-unaware basecallers or alterations in raw ionic currents and dwell times. In general, time-consuming and computationally intensive pre-processing steps are required, including raw signals basecalling by modification-unaware software, re-squiggling by aligning the ionic current data to the reference sequence, and modification detection by additional *ad hoc* models [6–8]. In our opinion, a critical bottleneck for epitranscriptome profiling is the lack of modification-aware basecallers that enable *ab initio* detection of canonical (A, C, G, U) and modified bases (m6A, m1A, I, and so on) in a single step, thus avoiding re-squiggling and the need for further models. A recent example is the *m*^{6}*ABasecaller* program, which allows the direct calling of m6A bases [9] in raw reads generated with the previous ONT chemistry. Moreover, ONT has released its modification-aware basecaller, Dorado, designed to process reads obtained with the latest sequencing chemistry. Dorado performs a first basecalling step i.e. insensitive to modifications, followed by the application of modification-specific models that detect and assign a probability to each modification (https://github.com/nanoporetech/dorado).

Translating raw ionic currents into base sequences might be carried out by transformer models that are used daily in speech-to-text translation algorithms to convert audio tracks into natural language texts and are characterised by a wide vocabulary. Such models, however, require adequate training datasets, which are currently missing for ONT dRNA. To face these challenges, we developed a computational pipeline based on two brand-new programs, NanoListener and NanoSpeech. NanoListener allows the creation of robust training datasets from ONT dRNA data coming from various organisms and synthetic constructs. NanoSpeech, instead, is a basecaller prototype implementing a transformer model with an expanded dictionary (A, C, G, U + mod bases), enabling the *ab initio* detection of canonical and modified bases in ONT dRNA sequences. By using real and synthetic constructs obtained by both the older (SQK-RNA002) and newer (SQK-RNA004) ONT chemistry, we prove that our transformer models are able to simultaneously call canonical and modified bases with an accuracy comparable to and sometimes higher than that of other state-of-the-art ONT basecalling programs.

Received: July 4, 2025. **Revised:** October 20, 2025. **Accepted:** December 16, 2025

Materials and Methods

Synthetic constructs

Synthetic dRNA reads from IVT sequences containing inosines in place of guanosines, named pureI, and their unmodified counterparts, named pureG, were obtained from Nguyen *et al.* [6]. PureI and pureG comprise three 1-kb-long sequences, designed to ensure that Gs were separated by at least eight nucleotides to avoid any interference between all the 81 different 5-mers containing Gs. Reads from four 2.5 kb long IVT sequences, named CC (abbreviation of 'curlcakes'), containing all 5-mer combinations of canonical bases, were obtained from Liu *et al.* [8]. To increase the diversity of unmodified contexts, additional IVT dRNA reads were used [10] (see Data Availability).

Given the low inosine density of pureI samples, an *E. coli* transcript from the AmpD gene (named coli-IVT), randomly chosen from known *E. coli* open reading frames with >20% Gs and no longer than 1000 bp and with no significant matches with human and mouse genomes, was *in vitro* transcribed by Trilink Biotechnologies (https://www.trilinkbiotech.com/). It consists of 672 bases and includes a polyA tail 120 nt long. The unmodified version was *in vitro* transcribed with a reaction mix of canonical ribonucleotides (ATP, CTP, GTP, UTP), while modified transcripts were synthesised, replacing GTP with ITP (ATP, CTP, UTP, ITP). This construct has a higher inosine density than the pureI sample (∼23% versus ∼10%) and includes kmers with multiple Is resembling a hyper-edited region.

Modified and unmodified dRNA reads from synthetic IVT sequences containing inosine, m1A, m6A, ac4C, m5C, hm5C, m5U, Psi, m1Psi and 2′-O-methylguanosine (Gm) were obtained from different public data sources and BioProjects (see Data Availability). The reads from synthetic constructs were generated using either the SQK-RNA002 or SQK-RNA004 chemistry (see Data Availability).

Real data

To increase the number of unmodified and inosine-modified contexts and make NanoSpeech models resilient to other epitranscriptomics modifications, additional data were obtained from publicly available transcriptomes of different organisms, including *E. coli* and mice [6] (see Data Availability). Specifically, we used dRNA raw reads described by Nguyen *et al.* [6], obtained from the total RNA of WT and ADAR1 E861A/ ADAR2 KO brains (which are completely A-to-I editing-deficient). Illumina RNAseq reads from the same mouse genotypes, described by Chalk *et al.* [11], were also used to create a grand-truth set of editing sites. Modifications-free dRNA data from IVT transcriptomes of human A549 and HeLa (two replicates) cell lines were also analysed [12] (see Data Availability).

Additionally, we sequenced in-house total RNA from yeast and human HEK293T cells. *Saccharomyces cerevisiae* W303-1B Wild Type (MATα ade2 leu2 his3 trp1 ura3) was cultivated in YPD-rich media at 30 °C for 6 h, and total RNA was extracted using the Presto™ Mini RNA Yeast Kit (GENEAID). Total RNA from HEK293T cells, including the WT cell line (hWT), the ADAR1 knockout (hKO) cell line, and the ADAR1 overexpressing (hOE) cell line, was extracted and purified as described in Fonzino *et al.* [13]. Illumina RNAseq data from the same samples were obtained from Fonzino *et al.* [13].

Direct RNA sequencing

Total RNAs (totRNA) from *S. cerevisiae* (supplied by Prof N. Guaragnella at UNIBA), HEK293T cell lines (supplied by Dr. R. Pecori and Dr A. Arnold at DKFZ) as well as coli-IVTs poly(A)RNAs samples were used for Nanopore libraries preparation with the SQK-RNA002—Direct RNA Sequencing Kit, following the protocol 'Direct RNA sequencing (SQK-RNA002) Version: DRS_9080_v2_revT_ 14Aug2019'.

For totRNA samples, the first two libraries were prepared from 500 ng of totRNA input, following the manufacturer's instructions. Subsequently, to increase the read number per sample, the totRNA input was increased up to 2 μg. For poly(A) RNA samples, the first library was prepared from 50 ng of poly(A) RNA input; for the second library prep, the input was increased to 150 ng.

Input RNA was concentrated using RNAClean XP (Beckman Coulter) beads 1.8X to reduce the sample volume, when necessary.

Each library was loaded on a single R9.4.1 Promethion flow cell using the Flow Cell Priming Kit (EXP-FLP002), and the sequencing run was carried out on the P2 Solo device. Sequencing runs were monitored through the ONT MinKNOW software using the fast basecalling modality and stopped after ∼72 h. Raw reads in FAST5 format were processed for downstream analyses.

Processing of Illumina RNAseq data and A-to-I RNA editing detection

Raw Illumina RNAseq data were quality-checked via FastQC and trimmed to remove adapters using fastp, as described by Fonzino *et al.* [14]. Trimmed and cleaned mouse reads were aligned onto the mm39 genome, while human reads were aligned onto the hg38 assembly by STAR [15]. Mouse and human reads were also aligned onto their reference transcriptomes and preprocessed by a custom Python script. Briefly, for each organism, RefSeq annotations from UCSC were analyzed to remove overlapping transcripts from adjacent loci and retain only the longer mature isoform per gene. Transcriptome alignments were carried out by BWA mem (v0.7.4) [16]. Output files were then filtered to remove unmapped reads and converted into coordinate-sorted BAM files using SAMtools v1.3.1 [17], as described in Fonzino *et al.* [14].

The REDItools [18, 19] package was used to detect RNA variants in both transcriptome and genome alignments. Lists of bona fide editing sites for humans and mice were obtained by comparing wild-type with ADAR knock-out samples, as previously described by Fonzino *et al.* [14]. Sites supported by at least 30 high-quality short reads, displaying an A-to-G editing level of at least 5% in the WT (or hOE for HEK293T cell lines) samples and showing no base change in the background knock-out samples were included in the ground-truth sets for downstream analyses. Lists of editing sites were further filtered to include only sites with editing evidence in all replicates (when available). For simplicity, only sites detected on the plus strand were provided in input to NanoListener for building the training datasets.

Pre-processing and mapping of dRNA reads

Raw fast5 files from ONT direct-RNA sequencing runs were basecalled by Guppy (version 5.0.11) using the following command line: guppy_basecaller -c rna_r9.4.1_70bps_hac.cfg -i [FAST5_Dir] -s [output_Path] -r – [fast5_out] -x 'cuda:auto.' For the preliminary test (Fig. 1G) on RNA004 reads, base-calling was performed using Dorado v.0.8.2 with rna004_130bps_sup@v5.1.0 model and the—emit-fastq flag to retrieve FASTQ files. The other RNA004 reads, derived from different modified and unmodified IVTs, were basecalled using Dorado v.1.0.2 with the rna004_130bps_sup@v5.2.0 model, requesting modifications via the four available models (totalling eight modifications). Basecalled reads passing the quality control step were merged into unique FASTQ files before the mapping. To increase the variability of the training dataset, failed reads from the coli-IVT samples were also used. Graphmap2

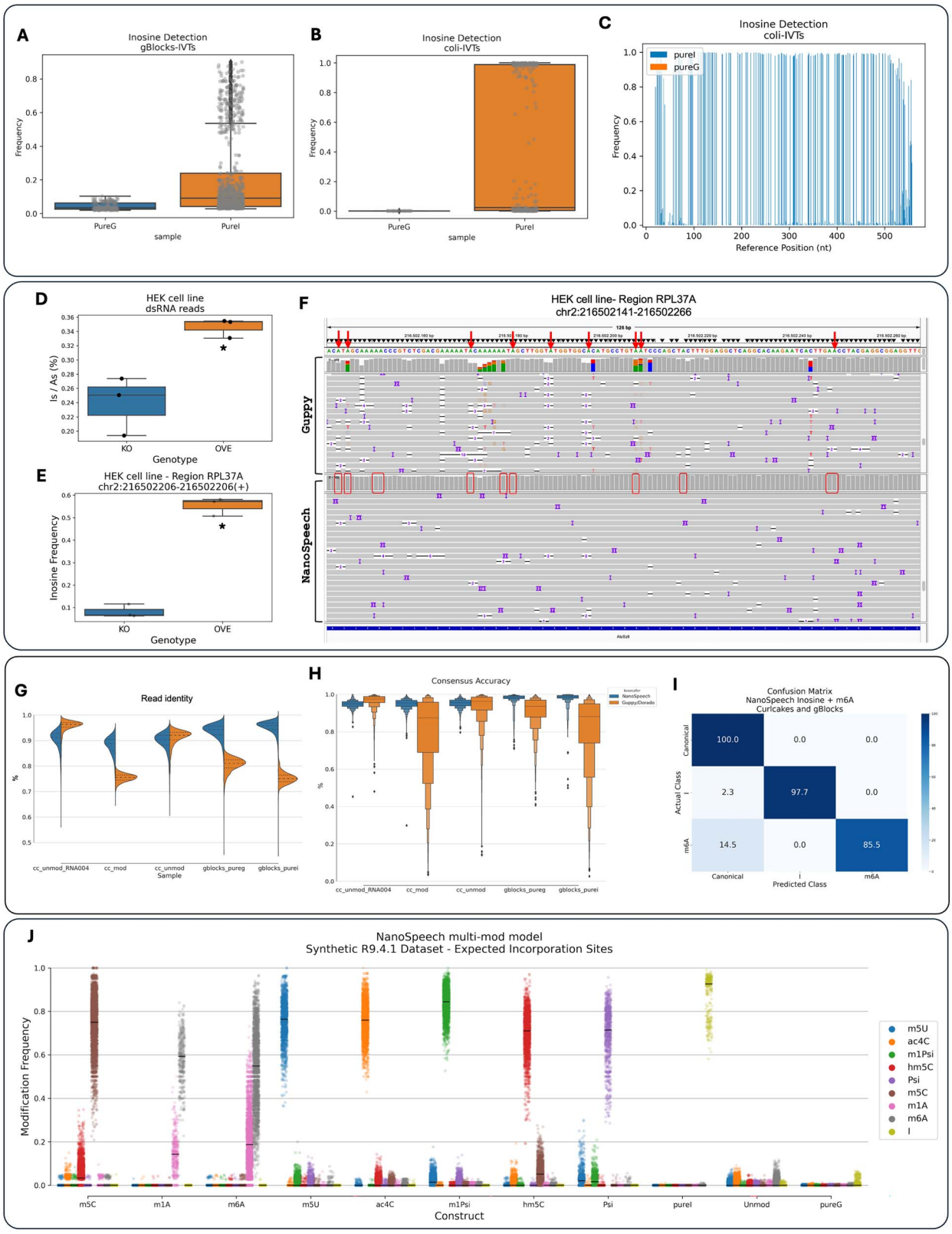

Figure 1. NanoSpeech *ab initio* detection of up to nine different modifications (RNA002 reads). (A) The NanoSpeech Inosine model (A, C, G, U and I), trained on synthetic IVTs and real-world multi-species reads, has a higher generalisation capacity and can accurately detect inosines in pureI reads while maintaining a low false-positive rate in pureG reads. (B) NanoSpeech successfully detects inosines in the high-density inosine coli-IVT sample at expected sites and (C) frequencies. (D) In a subset of human (HEK293) reads overlapping putative dsRNAs, NanoSpeech predicted more inosines in ADAR1 hOE than in hKO samples. (E, F) An example of NanoSpeech capabilities in detecting inosines in real data is reported. Specifically, alignment profiles of NanoSpeech reads (F, top) in a 125-nt long region falling in an Alu sequence of the RPL37A locus are shown, along with reads basecalled by Guppy (F, bottom). RNA editing sites confirmed by Illumina are indicated by red arrows, while NanoSpeech-detected inosines are depicted as red squares. NanoSpeech reads appear less noisy than Guppy reads and can efficiently predict editing sites, even in tricky homopolymeric regions. (G–I) A NanoSpeech model with an extended dictionary (m6A and I) was trained on IVT synthetic reads and used to basecall unseen unmodified or modified

v0.6.5 [20] was employed to perform less stringent alignments of pureI and pureG IVT reads using the command line: graphmap2 -x rnaseq -t 3 -r [REF.fa] -d [READS.fastq] -o [OUT.sam]. ColiIVT reads basecalled with Guppy were mapped by minimap2, decreasing the k parameter to 5 to increase the alignment rate of modified reads to the reference sequence, by the command line: minimap2 -t 5 -ax map-ont -k 5 —for-only —secondary = no —MD [REF.fa] [READS.fastq]. All other RNA002 dRNA reads were aligned by minimap2 [21] against the corresponding reference sequences using the default k = 14. Spliced mapping was activated by the -ax splice -uf option in minimap2. RNA004 reads basecalled by Dorado were instead mapped using the Dorado aligner program with default presets. The Modkit (v0.5.0) pileup command with its default thresholding strategy was used to extract the aggregated modification counts produced by Dorado for every position. To compute the modification ratios, these counts were normalised for the sequencing depth. To perform m6A-aware basecalling using m6Abasecaller [9], Guppy v.6.1.2 was used to run the template_rna_r9.4.1_70bps_m6A_hac.jsn custom model, which was downloaded from https://github.com/novoalab/m6ABasecaller repository, along with its rna_r9.4.1_70bps_m6A_hac.cfg configuration file. Subsequently, modPhred v.1.0d was exploited to align the basecalled reads and retrieve per-position modification frequencies from output bedMethyl files. SAM to BAM conversion, filtering, and sorting were performed with SAMtools [17]. Reads in the pod5 format were converted to fast5 by the ONT pod5 utilities before their analysis, when required.

Building training datasets by NanoListener

A strong limitation of existing models for basecalling dRNA reads is the lack of reliable training datasets as well as dedicated programs to build custom training datasets. For example, tools like taiyaki (developed by ONT) are deprecated, difficult to use, and dependent on the sequencing chemistry and kit. In addition, training datasets for already implemented models are not always available, raising questions about the reproducibility of results and limiting the development of more effective models. To overcome the existing problems, we have developed NanoListener, a program that allows the extraction of modified and unmodified current chunks from input dRNA reads for building robust training datasets. Specifically, dRNA reads are basecalled by modification-unaware software (such as Guppy or Dorado), aligned onto the corresponding reference sequence, and re-squiggled by f5c eventalign [22] (Fig. 2B). Then, NanoListener leverages the f5c eventalign output to anchor current chunks of variable length on the reference sequence and extract the corresponding reference kmers. Raw currents of extracted chunks are retrieved from fast5/pod5 files and projected onto the picoAmpere (pA) scale using the range, digitisation, and offset variables.

To deal with fluctuations in the translocation speed of DNA/RNA strands, NanoListener applies data augmentation procedures on individual reads, increasing the variability of extracted chunks/kmers couples (Supplementary Figs S14A–C). Indeed, two identical transcripts can produce current signals completely different in terms of translocation time and, thus, duration per event (the kmer context can change in the pore pocket). In particular, NanoListener performs an in-silico 'simulated' randomer strategy by which current chunks of variable lengths and offsets are sampled from the beginning of the raw fast5/pod5 signal. The number of random chunks per read is automatically computed and balanced according to the requested length range of chunks, allowing the traversing of the same signal multiple times and the generation of a larger number of examples (Fig. 2B).

NanoListener can take into account per-transcript annotations of RNA modifications and mark output kmers of the corresponding chunks. In synthetic constructs, the extraction of chunks linked to modified kmers is quite trivial because all reads are expected to be modified at the same positions. In reads from real data, instead, modified and unmodified bases coexist at a given position with different ratios, challenging the extraction of modified kmers and the creation of reliable training datasets. To overcome this issue, an anomaly detection approach based on IsolationForest (iForest) models was implemented. Given a couple of control/condition samples (CTRL/COND), where CTRL could be an IVT transcriptome or a knock-out sample, and a list of bona fide modified positions, basecalling features, such as base qualities and the number of mismatches and indels in 3 nt-long windows flanking each position of the bona fide list in the CTRL sample were extracted for training an iForest model. Then, it is applied to the COND sample to classify modified reads at bona fide positions as anomalies (Fig. 2C). Once modified and unmodified positions have been detected at the per-read level, NanoListener can collect current chunks and the corresponding kmers with canonical or modified bases (Fig. 2C).

To build the inosine dataset, NanoListener was launched on hWT, hOE, and hKO samples from human HEK293T cells and mice brains, using lists of bona fide editing sites obtained by Illumina RNAseq data from the same samples, as described above. All human and murine dRNA reads were aligned onto the corresponding reference transcriptomes by minimap2 (see above for more details). Aligned reads were further filtered by specific NanoListener routines that take into account the minimum number of reads per transcript, the average base quality per read, the average mapping quality per read, the read length, and the minimum coverage and depth. NanoListener was also applied to modified and unmodified dRNA reads from all synthetic constructs, including reads produced by the latest sequencing SQK-RNA004 kit. It is worth noting that NanoListener can easily handle dRNA reads from old and new sequencing kits and can be quickly adapted to future ONT releases.

In the present paper, NanoListener was used to create an inosine dataset comprising chunks and kmers from synthetic and real reads from yeast, *E. coli*, mice, and humans (see Data Availability). An additional inosine dataset was created, including chunks and kmers from the coliIVT sample. The m6A-I dataset, which contains chunks and kmers for inosines and m6A modifi-

reads, containing either m6A or inosine. Read identity (G) and consensus accuracy (H) for samples with and without m6A or I have been calculated to compare NanoSpeech with Guppy. A preliminary comparison between NanoSpeech and Dorado on RNA004 reads (cc_unmod_RNA004) is reported to assess the compatibility of our system with the latest ONT pore and chemistry. In (I) is shown a confusion matrix for modification predictions from the NanoSpeech m6A-I-model onto curlcakes (m6A) and gBlocks (inosine) IVT reads. (J) Our systems were tested in a more challenging scenario, training a NanoSpeech model with an expanded vocabulary of up to nine modified bases, using unmodified and fully modified (I, m1A, m6A, m5C, hm5C, ac4C, m5U, Psi, and m1Psi) synthetic IVT molecules. This model accurately identified all the modifications with very few cross-misclassifications. Specifically, a miscalling of the isomers m1A and m6A was found, likely due to several technical or sample-related issues as discussed in the main text (black lines indicate the median of distributions). (* $P < .05$ with unpaired two-tailed t-test).

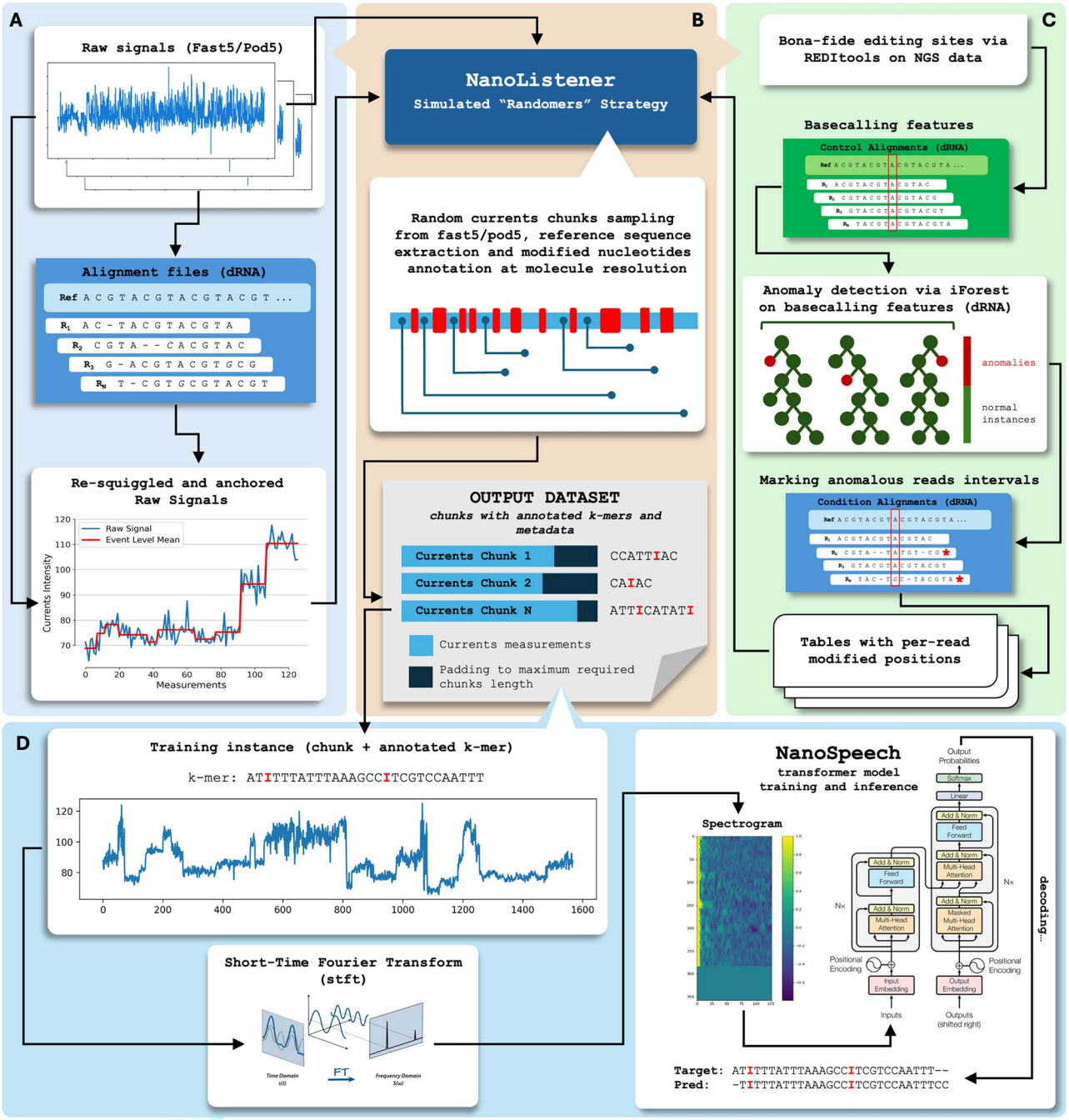

Figure 2. Overview of NanoListener and NanoSpeech. (A) Raw signals in fast5/pod5 files are basecalled via a modification-unaware tool and mapped against reference sequences. Alignments are filtered, and ionic current signals are split into events and re-squiggled onto the reference using f5c eventalign to obtain their precise mapping position. (B) NanoListener uses re-squiggled signals to retrieve context information from eventalign tables. It extracts random chunks of electric measurements from fast5, trying to avoid perturbed alignment regions (shown in red) due to modified bases. For each read, a variable number of pA-scaled current chunks is dynamically extracted, along with their associated k-mer annotated for modified bases, when needed. A padding is added to make both of them uniform in length. (C) Providing a set of bona fide modified sites, NanoListener extracts positional and per-read information from various sources, such as basecalling features of IVT or KO samples, which are used to train an isolation Forest model. Such iForest model was used to identify kmers with modified bases and their positions, creating a training dataset that embeds this metadata and feeds it to NanoListener. (D) Annotated, balanced and filtered NanoListener datasets were used to train NanoSpeech models with a user-preferred vocabulary. The current chunks were pre-processed via STFT into padded spectrograms, and several NanoSpeech models, through a classical encoder-decoder transformer, were trained to predict annotated output kmers for every input spectrogram. In inference mode, NanoSpeech decodes per-base probabilities into nucleotide sequences and prints out Fasta/Fastq files. In downstream procedures, position mapping with a per-read resolution of all the basecalled modified bases can be easily retrieved using accessory NanoSpeech scripts and aggregated to the genome-space level.

cations, was built using synthetic data (see Data Availability). The datasets for multiple modifications were created using synthetic data and sequences with either SQK-RNA002 or SQK-RNA004 chemistry (see Data Availability). To evaluate the compatibility of NanoListener with Uncalled4 [23], a recent re-squiggling software alternative to the f5c program, we created a training dataset from RNA004 unmodified IVT molecules using Uncalled4 eventalign tables by the following command: uncalled4 align —ref $GENOME —reads $FAST5 —bam-in $BAM —eventalign-out $EVOUT —processes 4 —read-filter $READSLIST —eventalign-flags print-read-names,signal-index,samples. In this way, a compatible f5c eventalign-like table was obtained. To note, NanoListener internally uses a summary table provided by the f5c eventalign to determine the exact location of each read within a number of fast5 files. Because of that, we used the summary table of the same run produced by f5c in combination with the eventalign table produced by Uncalled4 to overcome this issue and generate a valid dataset to train and validate NanoSpeech models. NanoListener implements a data augmentation strategy to increase the size of the training dataset and the diversity of examples from the same set of reads. To demonstrate its robustness, we simulated its absence by depleting all the overlapping chunks/k-mers via downstream filtering, using a pre-built dataset from RNA004 IVT unmodified synthetic reads. Then, NanoSpeech models were trained, evaluated and compared on datasets with or without data augmentation, fixing the number of epochs to 20 for the augmented dataset and increasing to 40 for the dataset without augmentation procedures (Supplementary Fig. S1).

NanoSpeech architecture, training, and inference

Training datasets generated by NanoListener were used to feed NanoSpeech models, which are based on an encoder-decoder Transformer architecture. Each extracted chunk of ionic currents in the pA scale was converted into a vector of M measurements c = [s1, s2, s3, . . . , sM] where curr_min_len < =M < =curr_max_len, as well as the corresponding kmer k = [y1, y2, y3, . . . , yL] of length L with nt_min_len < =L < =nt_max_len. The y vocabulary of k output vectors was expanded to include Nn non-canonical bases other than the four canonical ribonucleotides in such a way y ∈ {A, C, G, U, N1, N2, . . . Nn, <, >, −}, where the last three tokens were used as start, stop and silence symbols, respectively. To make all input vectors uniform in length, 0 values or silence tokens ('−') were added to the current and kmer vectors until the curr_max_len and nt_max_len were reached, respectively. Current vectors were converted into time-frequency representations (spectrograms, Fig. 2D) using Short-time Fourier Transform (STFT) and embedded in three Convolutional Neural Networks (CNN) layers interposed with ReLu activation functions. Inputs were then fed to encoder modules composed of several multi-headed attention layers, followed by dense DNN modules regularised with dropout and normalisation layers. Decoder modules received as inputs both shifted output sequences, and encoder outputs tried to generate the target kmer. Output vectors were encoded in numerical vectors where every y token was converted as a multilevel factor variable e such that f(y) = e and e ranged from 0 to N (where N is equal to 7 for the Inosine model, eight for the m6A-I model, 15 for the RNA002 multi-modification model and 18 for the RNA004 multi-modification model). For training purposes, a categorical cross-entropy was used in combination with an Adam optimiser, featuring a custom schedule for the learning rate routine over epochs. A dataset subset (3%) was used as a validation set to evaluate generalisation capacity after every training loop. At the end of each epoch, the expected output and predicted kmers were aligned using pairwise alignment via the Bio.pairwise2 module of the BioPython package [24] for visual inspection.

Incremental learning was employed to train resilient NanoSpeech models. To increase the number of chunks from synthetic modified IVTs, failed reads (i.e. reads that could not be basecalled by Guppy) were basecalled by pre-trained NanoSpeech models and processed by NanoListener. To improve the model configuration, a variety of combinations of chunk lengths and NanoSpeech models from different epochs and training datasets were explored (Supplementary Fig. S14). Every model and configuration was tested on a subset of independent reads comparing NanoSpeech and Guppy/Dorado alignments. Since raw-signal event length can vary significantly due to semi-stochastic fluctuations in translocation speed, different padding strategies have been applied on both currents chunks and output k-mers trying to take into account the known fixed sampling frequencies and the expected translocation speeds for the two different chemistries tested in this study (70 bps / 3 kHz and 130 bps / 4 kHz for RNA002 and RNA004 kits, respectively). For the RNA002 inosine-aware model trained on the multi-species dataset, chunks with length spanning from 500 to 5000 measurements were extracted using NanoListener and a padding was added to all of these, when required, in order to reach 5000 values, while the corresponding k-mers were padded to a very high maximum length value of 250 bases/token (encoding these as previously described). The best-trained NanoSpeech model was then used in inference mode to extract current chunks from fast5 files, each with a fixed length of 3000 measurements. These chunks were padded to the maximum value seen during training procedures, which was, in this case, 5000. In this way, given the fixed input length of current measurement to be basecalled, 250 bases should represent a reasonable value to handle on one hand, very 'long' events, which should be accountable for an output kmer of just (at least) a single base and, on the other hand, putative multiple very 'short' events. Analogously, for RNA002 2- and 9-modifications model prototypes, NanoListener datasets were created with chunks spanning from 1300 to a maximum of 1900 current measurements. In inference mode, a fixed length of 1440 (padded to 1900) samples was used, while the basecalled predicted k-mers were padded to 70 nucleotides/tokens. This should be appropriate for most cases, as it aims to decrease the complexity of the problem and considers the shorter current chunks used. For RNA004 models, different chunking strategies were explored as shown in Supplementary Fig. S14. For the training of NanoSpeech models, a minimalist configuration was attempted using a NanoListener dataset with chunks having minimum and maximum lengths of 800 and 1300 current measurements, respectively. In this case, all the raw signal chunks were also padded to 1300 for both training and inference steps. During inference mode, the extracted chunks from fast5 (or converted pod5) files were composed of contiguous signals of 945 samples with corresponding basecalled output k-mers padded to a maximum length of 65 nucleotides.

NanoSpeech has been designed on top of the standard Encoder-Decoder Transformer example code from the Keras open-source library (https://keras.io/examples) with several tweaks to deal with input current features, steps to generate outputs, model structure (number of CNN and dense layers, encoders, decoders, multi-head attention layers, and so on) and the size and composition of the output dictionary. Additional routines were implemented to calculate Phred-like quality scores from 'logits' produced by the decoder module and to print out sequences in FAS-

TA/FASTQ format with embedded modified bases. Keras2 (https://keras.io/) and Tensorflow 2.13 [25] libraries were used in training and inference scripts.

Once trained on a specific dataset, NanoSpeech performs the basecalling on input reads of fast5/pod5 (converted) files. The whole current signal of each read is split into contiguous chunks of fixed length, and inferred kmers (trimmed by the stop '>' symbol) are concatenated in a final sequence. For modified bases, the canonical version is included in the output read, and an index corresponding to the exact sequence position is reported in the header of the final FASTA/FASTQ file. A utility script is also provided to retrieve the position of modified bases (based on the MD tag of the BAM file) after mapping onto the reference sequence.

All computations (training and inference) were carried out on a CentOS 6 server equipped with 32 CPUs, 512 GB of RAM, and a GPU Nvidia A100–40/80 GB.

The NanoSpeech models used in this paper are freely available on the GitHub page (see Code and Data Availability).

Evaluation metrics

NanoSpeech performance on canonical bases and mappability of basecalled reads were assessed on read subsets from all the experimental groups and compared to Guppy. Both NanoSpeech and Guppy basecalled reads were mapped with minimap2 onto the corresponding reference sequences using default parameters (see above). Percentages of unmapped reads were retrieved via the Pysam module (https://pysam.readthedocs.io/en/latest/), counting reads with the bitwise flag equal to 4 within each generated BAM file. CIGAR strings of aligned reads were parsed to compute detailed alignment metrics using the following formulas:

- Alignment: M + I + D
- Deletions: D
- Insertions: I
- Matches = M - (NM - (I + D))
- Percentage Identity = Matches / Alignment

where M = Matches+Mismatches, D = Deletions, I = Insertions, according to the CIGAR nomenclature of the SAM format. NM, instead, is the SAM tag indicating the number of mismatches.

The consensus accuracy was computed by analysing BAM files with the Pysam module, focusing on all the well-covered reference positions, and extracting the percentage of bases matching the reference on the total amount of mapped reads per site.

Model accuracy, sensitivity, specificity, f1-score, AUC score, and confusion matrices were calculated as described in Fonzino *et al.* [13].

Abbreviations for RNA modifications

The following abbreviations for RNA modifications were used across the manuscript: m1A 1-methyladenosine, m6A N6-methyladenosine, ac4C N4-acetylcytidine, m5C 5-methylcytidine, hm5C 5-hydroxymethylcytidine, m5U 5-methyluridine, I inosine, Psi pseudouridine, m1Psi 1-methylpseudouridine and Gm 2′-O-methylguanosine. Specific chemical properties of the above RNA modifications can be found in the MODOMIC database [1].

Results and discussion

Basecalling of inosine-rich reads

The feasibility of our approach was initially proved by building a training dataset for calling inosines in dRNA reads owing to their challenging identification. In eukaryotic transcriptomes, especially in humans, inosines are truly pervasive due to the hydrolytic deamination of adenosines by the ADAR family of enzymes [26–28], a process referred to as RNA editing. Using available modified (pure*I* RNA samples embedding inosines in place of guanosines) and unmodified (pure*G* RNA samples without any inosine) *in vitro* transcribed (IVT) synthetic sequences [6], a first training dataset was created by NanoListener, partitioning raw current intensities into discrete events by re-squiggling dRNA reads (using f5c [22]) and extracting current chunks corresponding to reference kmers (Fig. 2A). Due to fluctuations in the translocation speed of RNA strands through the pore, two identical transcripts can generate different current signals in terms of event duration (the kmer context can change in the pore pocket). Consequently, NanoListener implements an *in silico* 'simulated' randomer strategy for data augmentation, traversing the same signal multiple times (Fig. 2B). Based on the requested length range of chunks, their random number for every read is automatically computed and balanced. The initial dataset was enriched with additional chunks of dRNA sequences from several organisms, including yeast, *Escherichia coli*, mice, and humans. Although NanoListener handles by default re-squiggled dRNA reads by f5c, it can also operate with dRNA reads re-squiggled by other tools, such as the recent Uncalled4 [23] software, without significant differences in downstream training and inference processes (Supplementary Fig. S1). On the contrary, data augmentation is a mandatory step to ensure high-quality training datasets and reliable basecalling (see Methods and Supplementary Fig. S1).

A-to-I editing does not exhibit an all-or-nothing effect, and modified and unmodified transcripts from the same gene locus coexist in different ratios, making it challenging to extract real examples with sequence contexts containing Is. To overcome this issue, we used two strategies: (i) we downloaded public Illumina and ONT dRNA reads from wild-type, ADAR and ADARB1 knockout mice brains (from SQK-RNA002 chemistry) [6], and (ii) we sequenced total RNA from wild-type (hWT), ADAR knockout (hKO), and ADAR overexpressing (hOE) HEK293T cells by the same two technologies (using the SQK-RNA002 chemistry for ONT).

Illumina sequencing was exploited as an orthogonal technique to obtain a set of *bona fide* A-to-I editing events comparing wild-type and knockout human and mouse samples by our REDItools software [18, 19]. Genomic positions with A-to-I editing evidence were provided to NanoListener for extracting modified and unmodified chunks from dRNA reads. Modified chunks were selected as anomalies using an iForest-based algorithm, which has been successfully employed to identify rare C-to-U editing events in human dRNA reads (Fig. 2C) [14]. On the whole, more than 9 million chunks were extracted and fed to NanoSpeech, implementing a standard encoder-decoder transformer architecture able to read input padded chunks of electric signals and print out kmers of variable lengths (Fig. 2D). In its inference mode, NanoSpeech performs the *ab initio* base calling of FAST5 dRNA reads and generates FASTA/FASTQ output files in which modified bases are embedded in the header lines (Supplementary Fig. S2). Once trained, the NanoSpeech model was applied to subsets of unseen dRNA reads from pure*G* and pure*I* samples as well as to independent samples without Inosines (Is), including a synthetic construct (CC sample) and two IVT transcriptomes from A549 and HeLa cell lines. Resulting basecalled reads were compared to Guppy basecalled reads and further analysed to evaluate the detection performance of Is. NanoSpeech reads showed much higher mappability than Guppy reads in inosine-rich sequences (from ∼40% to more than ∼90%).

Inosines were detected with high accuracy in the pure*I* sample (accuracy >96%, f1-score > 95%, and AUC ~1.0) and only a negligible number of false negatives in samples lacking Is were found (Fig. 1A, Supplementary Fig. S3, S4). Since A-to-I editing appears in clusters, multiple Is may occur in the same and adjacent kmers, mostly hampering the basecalling of modification-unaware tools. To prove the power of our NanoSpeech-based approach, an *E.coli* transcript (from *AmpD* gene with no significant matches with mouse and human genomes) was *in vitro* transcribed both in its native form and replacing Gs with Is, likewise the pure*G* and pure*I* samples, but displaying additional inosine contexts resembling a hyper-edited region. Guppy basecalling on the I-rich sample was quite challenging, resulting in only a limited number of reads, which were used to extract random chunks and extend the NanoSpeech model. Notably, after the training, the NanoSpeech basecalling of unseen sequences of the I-rich sample yielded a higher number of correctly aligned reads than Guppy (>99%) (Supplementary Figs. S5, S6). Additionally, NanoSpeech detected Is at the expected reference positions with an accuracy of ~96%, an F1-score of ~95%, and a few false negatives (Fig. 1B and C, Supplementary Fig. S7).

NanoSpeech performance was also assessed on non-synthetic datasets using unseen hWT, hKO, hOE HEK293T dRNA reads, focusing on reads overlapping genomic loci harbouring predicted double-strand RNAs (dsRNA) deposited in the REDIportal [28] database and expected to be targeted by ADARs. Although not extensively trained in humans, NanoSpeech interestingly detected more Is in the hOE than in hKO samples, and 92% of identified Is in hOE samples were also present in REDIportal as known A-to-I events (Figs 1D–F). In addition, per-site inosine levels in hOE samples were well correlated (Spearman's $P < .001$) with those detected by Illumina.

Basecalling of multiple modifications

Since our NanoSpeech prototype is based on a transformer architecture used in speech-to-text translation, we tested a model capable of handling two different RNA modifications simultaneously using synthetic IVT datasets. NanoListener was, therefore, used to extract random chunks from unmodified and modified dRNA reads containing I or m6A, respectively. Only reads covering at least 70% of the reference sequences were used to maximise the heterogeneity of the kmer space for the training dataset. The new m6A-I-aware NanoSpeech model was then applied to basecall *ab initio* unseen dRNA reads. The extended nucleotide vocabulary (A, C, G, U, I, m6A) didn't affect the model performance. NanoSpeech reads displayed a higher per-read identity (Fig. 1G) and consensus accuracy (Fig. 1H) than Guppy-called reads in I- and m6A-rich sequences. NanoSpeech and Guppy's metrics were, instead, comparable in unmodified dRNA reads. The m6A-I-aware model achieved a precision of at least 0.98 for each class of sites, canonical, inosine, and m6A, and a recall of ~0.99 with a low false positive rate (Fig. 1I, Supplementary Fig. S8). Interestingly, no misclassification between I and m6A was found, demonstrating the model's capability to distinguish these two modified adenosines from each other and the canonical. We then compared the m6A-I NanoSpeech model with the m6Abasecaller [9] on raw reads from unmodified and fully modified (with m6A) synthetic constructs. NanoSpeech called a higher fraction of m6A with expected stoichiometry than m6Abasecaller (Supplementary Fig. S9). Moreover, alignments of reads basecalled by m6Abasecaller overlapped with those generated by Guppy but appeared noisier than the alignments of reads basecalled by NanoSpeech (Supplementary Fig. S9).

To demonstrate the power and flexibility of our approach, a transformer model capable of the *ab initio* concomitant detection of up to nine different RNA modifications was set up. Unmodified and modified reads (from SQK-RNA002 chemistry) incorporating m1A, m6A, ac4C, m5C, hm5C, m5U, I, Psi, and m1Psi were analysed by NanoListener to create an enlarged training dataset for an extended NanoSpeech multi-model. Once applied to unseen IVT dRNA reads, our multi-model transformer was able to detect the expected modification for each sample, maintaining a low false positive rate. However, the detection accuracy appeared dependent on the chemical nature of the modification (Fig. 1J). For example, m1A was erroneously miscalled as its chemical isomer m6A. Interestingly, when the m6A-I model was used to basecall fully modified reads incorporating m1A, we found a high percentage of m6A (Supplementary Fig. S10), suggesting that the original sample may contain a mixture of m1A and m6A. On the other hand, current values for m1A and m6A contexts at DRACH containing k-mers (preferred by m6A writer enzymes) were not significantly different, in most cases (Fig. 1J, Supplementary Fig. S11, S12). The miscalling of m1A and m6A could be due to multiple factors such as: technical limitations in the ability to distinguish chemically similar bases, the lack of streamlined models able to capture nearly undetectable current changes linked to highly similar chemical structures, or intrinsic errors in the modified samples (limited number of reads, low quality of input RNAs, contaminations and different experimental conditions used to generate IVTs). According to our findings, intrinsic errors in the m1A sample could have affected the training of the multi-modification model. Indeed, it is well known that m1A can be converted into m6A via the Dimroth rearrangement, a chemical transformation occurring under heat and alkaline conditions [29]. Moreover, when both the m6A-I model and the multi-modification model were used to basecall reads of a synthetic construct fully modified with m6A, the correlation of detected m6A bases was only 0.451. By considering both m6A and m1A in the multi-modification model, the r-value rose to 0.709 (Supplementary Fig. S10). On the contrary, when both models were applied to a synthetic construct fully modified with inosines, their correlation was 0.996 (Supplementary Fig. S10). Such results suggest potential bias in the training dataset of the multi-modification model for m1A contexts. Since the presence of m6A in the m1A sample cannot be assayed and validated by orthogonal methods, technical and/or methodological limitations due to highly similar bases cannot be excluded. Nonetheless, the NanoSpeech multi-model was always able to distinguish the canonical base from the modified counterpart (i.e. m6A versus A or m5U versus U and so on) (Fig. 1J). In addition, the multi-model improved the mappability of modified reads, sometimes dramatically, as for the Psi reads basecalled by Guppy and NanoSpeech (Supplementary Fig. S13). The latter allowed the alignment of 92% of reads more than the former (504/538 versus 8/538).

Basecalling of ONT RNA004 reads

NanoListener and NanoSpeech are independent of the specific ONT chemistry, meaning that they can also process reads generated with the newer SQK-RNA004 chemistry. To demonstrate the flexibility and adaptability of our tools to the new pore and chemistry, we downloaded RNA004 raw synthetic data from unmodified and modified samples, including up to eight different RNA modifications. Specifically, we used four short sets of oligos

Figure 3. *Ab initio* multi-modification detection in RNA004 IVT reads and comparison with other tools. We evaluated the flexibility and adaptability of NanoListener and NanoSpeech in performing *ab initio* detection of multiple modifications in reads from the latest ONT chemistry using a single transformer model. IVT synthetic constructs with or without modifications were retrieved from different public data sources (see Data Availability for further details) (cc_∗: Long IVT curlcakes [9], ONT_∗: Short IVT oligos from ONT open data; Validation_∗: Short IVT oligos with or without one/two modifications). Both curlcakes and ONT reads were analysed by NanoListener to build an initial dataset, which was used to train a NanoSpeech model for the concomitant identification of eight different modifications. Alignment profiles of NanoSpeech on unseen reads were compared to Dorado in terms of percentage of read-identity, including (A) or excluding (B) insertions, and consensus accuracy (C). The NanoSpeech multi-modification model was able to call and retrieve frequencies at expected incorporation sites, which were then aggregated and displayed as dot-plots (D). (E–K) Predicted

(with a length of 100 bases, N = 6) incorporating inosine, m5C, m6A and pU at defined positions, as well as seven sets of fully modified IVT constructs (with a mean length of 2533.75 ± 173.11 SD bases, N = 4) including ac4C, hm5C, m1Psi, m5C, m5U, m6A and Psi. NanoListener was successfully used to create a training dataset for a NanoSpeech multimodel, which was subsequently applied to previously unseen dRNA reads. Basecalling results were compared to those obtained using Dorado, where RNA modifications were detected in post-processing with the ModKit tool. As shown in Fig. 3A, Dorado reads displayed a slightly higher per-read identity than NanoSpeech reads; however, negligible differences were observed when insertions were excluded (Fig. 3B). As a prototype, NanoSpeech does not yet implement an optimised algorithm to merge adjacent chunks during the inference step, and, thus, a higher insertion rate is expected. Conversely, NanoSpeech and Dorado showed similar performance in terms of consensus accuracy (Fig. 3C), demonstrating the basecalling effectiveness of NanoSpeech. Major or minor differences in per-read identity and consensus accuracy depended on chemical modification. For example, NanoSpeech outperformed Dorado in reads containing ac4C or m1Psi, while Dorado yielded better results with reads incorporating m5C or m6A. It is worth noting that Dorado detects modifications applying multiple internal models and post-processing with ModKit, which aggregates modification counts across transcriptomic positions and applies sample-specific thresholds for calling modified bases. In contrast, NanoSpeech detects modifications in a single step, simultaneously with canonical bases, using a unified model trained for multiple modification types.

Consistent with previous results with the older chemistry, NanoSpeech was able to identify several RNA modifications *ab initio* (Fig. 3D), albeit with some limitations, particularly for reads containing ac4C or hm5C. As previously discussed regarding the miscalling of m6A and m1A, these limitations may arise from multiple factors, including the read number and quality, the size of the training dataset, the chemical similarity among RNA modifications, and technical and/or methodological constraints.

Using the same synthetic reads, we also assessed the ability of NanoSpeech and Dorado in basecalling modifications and estimating their stoichiometry. While results were comparable for short and partially modified oligos (Figs 3E–H), NanoSpeech outperformed Dorado in fully modified constructs containing m6A, m5C and Psi (Figs 3I–K), demonstrating once again the strength of our approach. Finally, we evaluated the NanoSpeech performance on synthetic RNA oligonucleotides containing unmodified bases and up to two different modifications within the same molecule, already used to test ModiDeC, a recently released classifier for RNA modifications in ONT native RNA reads [30]. Focusing on unmodified reads and reads containing 2′-O-methylguanosine (Gm) and m6A, or inosine and Gm, at two specific positions, the NanoSpeech multimodel (re-trained to accommodate contexts of these new oligos) was able to successfully detect Gm and m6A, or inosine and Gm, at the expected positions and with the expected stoichiometry in completely unseen reads (Figs 3L–N).

Conclusions

In summary, our findings strongly indicate that transformer models with expanded vocabularies are effective in identifying *ab initio* multiple modifications in dRNA reads. In the meantime, our NanoListener allows the extraction of current chunks and kmers from dRNA reads produced by whatever ONT sequencing chemistry and kit, enabling the creation of custom training datasets for a large variety of applications.

Although primarily tested on synthetic constructs as a proof of concept, NanoSpeech models demonstrated accuracy comparable to that of state-of-the-art ONT software in both basecalling reads and RNA modification detection. NanoSpeech multi-modification models can distinguish canonical from modified bases without additional post-processing steps (such as the use of ModiKit or similar tools), albeit with some limitations. Indeed, highly similar chemical structures or intrinsic errors in the modified samples, such as the number of available reads and their quality, contaminations and sometimes unknown experimental conditions to generate IVTs, hamper the generation of robust training sets leading to inference artefacts or 'hallucinations.' Streamlined NanoSpeech models enabling the detection of highly similar chemical structures are currently under investigation and development. Nonetheless, several critical challenges remain to be addressed, including the often questionable quality of public data for RNA modifications and the lack of reliable benchmarks for training and validation, both of which impair accurate modification detection in real-world data. Moreover, the unavailability of training datasets used by current basecalling tools limits the possibility of unbiased and reproducible performance comparisons.

Despite the challenges and the need for further optimisations, NanoListener and NanoSpeech represent a novel and very promising methodological framework for incoming nanopore-based technologies, addressing direct protein sequencing and calling of canonical and modified amino acids.

Key Points

- Current bioinformatics tools for profiling RNA modifications using ONT dRNA sequencing rely on modification-unaware software and require multiple modification-specific learning steps.
- To improve the identification of epitranscriptomic changes, we have developed two brand-new programs, NanoListener and NanoSpeech.
- NanoListener implements a simulated randomers strategy to build robust training datasets from ONT dRNA

modification ratios in partially or fully modified molecules for NanoSpeech and Dorado. Four of these (E–H) consisted of short oligos incorporating m6As (E), m5Cs (F), pUs (G) or inosines (H) in about five positions per construct, while long fully modified IVT synthetic constructs are depicted in panels I–K. (L and M) An extended NanoSpeech model was trained using reads from oligos incorporating two different modifications on the same molecule at two specific positions, Gm-m6A (L) and Inosine-Gm (M). Unmodified reads were also used (N). NanoSpeech correctly detected all modifications at expected positions and with expected stoichiometry, with a very low false positive rate. (A and B) Boxen plots are used to represent distributions of percentages of identity computed for all analysed reads, (C) Box plots represent distributions of consensus accuracies computed across all well-covered reference positions; (D–N) Each dot represents the modification frequency detected for a given site, which was computed aggregating predictions at per-read level; (E–K) The same distributions are summarised by box plots; horizontal black lines represent the median of distributions; vertical dotted-red lines depict incorporation sites across IVT oligo expected to have two distinct modifications per-molecule).

sequencing datasets, regardless of the chemistry, fostering the development of a new generation of ONT basecallers.
- NanoSpeech is a modification-aware basecaller for the *ab initio* simultaneous detection of multiple modified bases in ONT dRNA reads, using a transformer model with *an* expanded vocabulary.
- Using synthetic and real datasets, NanoSpeech can profile up to nine different modifications *ab initio* using a unique model.

Acknowledgements

The authors thank D.A. Silvestris for suggestions about the detection of RNA editing sites from Illumina HEK293T cells, P. D'Addabbo for providing the list of putative dsRNAs, E. Filomena for solving some technical issues, N. Guaragnella for supplying yeast RNA, and R. Pecori and A. Arnold for providing HEK293T RNA.

Authors are grateful to the following National Research Centres: 'High Performance Computing, Big Data, and Quantum Computing' (Project no. CN_00000013) and 'Gene Therapy and Drugs based on RNA Technology' (Project no. CN_00000041); and Extended Partnerships: MNESYS (Project no. PE_0000006) and Age-It (Project no. PE_00000015). The work was also supported by Life Science Hub Regione Puglia (LSH-Puglia, T4-AN-01 H93C22000560003) and INNOVA - Italian network of excellence for advanced diagnosis (PNC-EJ-2022-23683266 PNC-HLS-DA) and by ELIXIR-IT through the empowering project ELIXIRNextGenIT (Grant Code IR0000010).

Author contributions

A.F., G.P., and E.P. conceived the study. G.P. and E.P. supervised the research. A.F. developed the NanoListener and NanoSpeech algorithms and performed the main data analyses. B.F. supervised the data analyses. G.V. and C.G. performed ONT sequencing. A.F. and E.P. wrote the paper with input from all other authors. All authors read and approved the final manuscript.

Supplementary data

Supplementary data is available at *Briefings in Bioinformatics* online.

Conflict of interest

None declared.

Funding

None declared.

Data availability

Unmodified pureG and modified pureI reads were downloaded from the BioProject PRJNA814100. Unmodified reads from constructs containing all 5-mer combinations of canonical bases were obtained from the BioProject PRJNA511582. Additional IVT dRNA reads were downloaded from BioProjects PRJNA497103 and PRJEB44511.

RNA002 IVT sequences containing m1A, m6A, ac4C, m5C, hm5C, m5U, Psi, and m1Psi were downloaded from the following BioProjects PRJEB73868, PRJEB67632, PRJNA548268, PRJNA549001 and PRJNA1050579. Modified and unmodified reads from synthetic IVT sequences generated by the latest RNA004 chemistry were obtained from different data sources. More in detail, unmodified and modified IVTs curlcakes were downloaded from BioProjects PRJEB73868 and PRJEB82528. Shorter unmodified and partially modified oligos, made freely available by ONT (https://epi2me.nanoporetech.com/rna-mod-validation-data/), were downloaded from the s3://ont-open-data/rna-modbase-validation_2025.03 AWS s3 object storage service. Recent ModiDeC [30] training and validation reads, comprising innovative unmodified and partially modified (single or a couple of modifications) oligos, were downloaded from the BioProject PRJEB88778.

E. Coli RNA002 dRNA reads were downloaded from SRA using the accession SRR18070402. Raw RNA002 dRNA reads of the mouse brain were downloaded from the BioProject PRJNA814100, while the corresponding Illumina reads were downloaded from the BioProject PRJNA546532. Raw RNA002 dRNA data of IVT transcriptomes from human A549 and HeLa (two replicates) cell lines were downloaded from SRA using the following accessions: SRR23950397, SRR23950400, and SRR23950399, respectively. Illumina reads from HEK293T cells were downloaded from the BioProject PRJNA1138417.

Modified and unmodified RNA002 reads from *E. coli* AmpD, as well as yeast and human (HEK293T) dRNA reads produced in this study, are available at the BioProject PRJNA1243074.

Training datasets generated by NanoListener are available upon request.

Code availability

NanoListener and NanoSpeech, along with models and auxiliary scripts, are available at the following GitHub pages https://github.com/F0nz0/NanoListener and https://github.com/F0nz0/NanoSpeech_basecaller, respectively.

References

1. Boccaletto P, Stefaniak F, Ray A, *et al.* Update. *Nucleic Acids Res* 2021;**50**:D231–5. https://doi.org/10.1093/nar/gkab1083.
2. Cerneckis J, Ming G-L, Song H, *et al.* The rise of epitranscriptomics: Recent developments and future directions. *Trends Pharmacol Sci* 2024;**45**:24–38. https://doi.org/10.1016/j.tips.2023.11.002.
3. Peer E, Rechavi G, Dominissini D. Epitranscriptomics: Regulation of mRNA metabolism through modifications. *Curr Opin Chem Biol* 2017;**41**:93–8. https://doi.org/10.1016/j.cbpa.2017.10.008.
4. Jain M, Abu-Shumays R, Olsen HE, *et al.* Advances in nanopore direct RNA sequencing. *Nat Methods* 2022;**19**:1160–4. https://doi.org/10.1038/s41592-022-01633-w.
5. Begik O, Mattick JS, Novoa EM. Exploring the epitranscriptome by native RNA sequencing. *RNA N Y N* 2022;**28**:1430–9. https://doi.org/10.1261/rna.079404.122.
6. Nguyen TA, Heng JWJ, Kaewsapsak P, *et al.* Direct identification of A-to-I editing sites with nanopore native RNA sequencing. *Nat Methods* 2022;**19**:833–44. https://doi.org/10.1038/s41592-022-01513-3.
7. Pagès-Gallego M, de Ridder J. Comprehensive benchmark and architectural analysis of deep learning models for nanopore

sequencing basecalling. *Genome Biol* 2023;**24**:71. https://doi.org/10.1186/s13059-023-02903-2.
8. Liu H, Begik O, Lucas MC, *et al.* Accurate detection of m6A RNA modifications in native RNA sequences. *Nat Commun* 2019;**10**:4079. https://doi.org/10.1038/s41467-019-11713-9.
9. Cruciani S, Delgado-Tejedor A, Pryszcz LP, *et al.* De novo basecalling of RNA modifications at single molecule and nucleotide resolution. *Genome Biol* 2025;**26**:38. https://doi.org/10.1186/s13059-025-03498-6.
10. Jenjaroenpun P, Wongsurawat T, Wadley TD, *et al.* Decoding the epitranscriptional landscape from native RNA sequences. *Nucleic Acids Res* 2021;**49**:e7. https://doi.org/10.1093/nar/gkaa620.
11. Chalk AM, Taylor S, Heraud-Farlow JE, *et al.* The majority of A-to-I RNA editing is not required for mammalian homeostasis. *Genome Biol* 2019;**20**:268. https://doi.org/10.1186/s13059-019-1873-2.
12. McCormick CA, Akeson S, Tavakoli S, *et al.* Multicellular, IVT-derived, unmodified human transcriptome for nanopore-direct RNA analysis. *BioRxiv Prepr Serv Biol* 2024.
13. Fonzino A, Mazzacuva PL, Handen A, *et al.* REDInet: A temporal convolutional network-based classifier for A-to-I RNA editing detection harnessing million known events. *Brief Bioinform* 2025;**26**:bbaf107. https://doi.org/10.1093/bib/bbaf107.
14. Fonzino A, Manzari C, Spadavecchia P, *et al.* Unraveling C-to-U RNA editing events from direct RNA sequencing. *RNA Biol* 2024;**21**:1–14. https://doi.org/10.1080/15476286.2023.2290843.
15. Dobin A, Davis CA, Schlesinger F, *et al.* STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics* 2013;**29**:15–21. https://doi.org/10.1093/bioinformatics/bts635.
16. Li H, Durbin R. Fast and accurate short read alignment with burrows-wheeler transform. *Bioinformatics* 2009;**25**:1754–60. https://doi.org/10.1093/bioinformatics/btp324.
17. Li H, Handsaker B, Wysoker A, *et al.* The sequence alignment/map format and SAMtools. *Bioinformatics* 2009;**25**:2078–9. https://doi.org/10.1093/bioinformatics/btp352.
18. Picardi E, Pesole G. REDItools: High-throughput RNA editing detection made easy. *Bioinformatics* 2013;**29**:1813–4. https://doi.org/10.1093/bioinformatics/btt287.
19. Lo Giudice C, Tangaro MA, Pesole G, *et al.* Investigating RNA editing in deep transcriptome datasets with REDItools and REDIportal. *Nat Protoc* 2020;**15**:1098–131. https://doi.org/10.1038/s41596-019-0279-7.
20. Sovic I, Sikic M, Wilm A, *et al.* Fast and sensitive mapping of nanopore sequencing reads with GraphMap. *Nat Commun* 2016;**7**:11307. https://doi.org/10.1038/ncomms11307.
21. Li H. Minimap2: Pairwise alignment for nucleotide sequences. *Bioinforma Oxf Engl* 2018;**34**:3094–100. https://doi.org/10.1093/bioinformatics/bty191.
22. Gamaarachchi H, Lam CW, Jayatilaka G, *et al.* GPU accelerated adaptive banded event alignment for rapid comparative nanopore signal analysis. *BMC Bioinformatics* 2020;**21**:343. https://doi.org/10.1186/s12859-020-03697-x.
23. Kovaka S, Hook PW, Jenike KM, *et al.* Uncalled4 improves nanopore DNA and RNA modification detection via fast and accurate signal alignment. *Nat Methods* 2025;**22**:681–91. https://doi.org/10.1038/s41592-025-02631-4.
24. Cock PJ, Antao T, Chang JT, *et al.* Biopython: Freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics* 2009;**25**:1422–3. https://doi.org/10.1093/bioinformatics/btp163.
25. Abadi M, Barham P, Chen J. *et al. TensorFlow: A System for Large-Scale Machine Learning*, 12th USENIX Symposium on Operating Systems Design and Implementation, 2016.
26. Tan MH, Li Q, Shanmugam R, *et al.* Dynamic landscape and regulation of RNA editing in mammals. *Nature* 2017;**550**:249–54. https://doi.org/10.1038/nature24041.
27. Picardi E, Manzari C, Mastropasqua F, *et al.* Profiling RNA editing in human tissues: Towards the inosinome atlas. *Sci Rep* 2015;**5**:14941. https://doi.org/10.1038/srep14941.
28. D'Addabbo P, Cohen-Fultheim R, Twersky I, *et al.* REDIportal: Toward an integrated view of the A-to-I editing. *Nucleic Acids Res* 2025;**53**:D233–42. https://doi.org/10.1093/nar/gkae1083.
29. Liu H, Zeng T, He C, *et al.* Development of mild chemical catalysis conditions for m1A-to-m6A rearrangement on RNA. *ACS Chem Biol* 2022;**17**:1334–42. https://doi.org/10.1021/acschembio.2c00178.
30. Alagna N, Mündnich S, Miedema J, *et al.* ModiDeC: A multi-RNA modification classifier for direct nanopore sequencing. *Nucleic Acids Res* 2025;**53**:gkaf673. https://doi.org/10.1093/nar/gkaf673.

Briefings in Bioinformatics, 2026, **27(1)**, bbaf709
https://doi.org/10.1093/bib/bbaf709
Case Study