Communications Biology

https://doi.org/10.1038/s42003-026-09757-x

Article in Press

Retentive Network promotes efficient RNA language modeling of long sequences

Received: 18 April 2025

Accepted: 16 February 2026

Cite this article as: Shen, Y., Cao, G., Hu, Y. *et al.* Retentive Network promotes efficient RNA language modeling of long sequences. *Commun Biol* (2026). https://doi.org/10.1038/s42003-026-09757-x

Yi Shen, Guangshuo Cao, Yueming Hu, Shilong Zhang, Jianghong Wu, Dijun Chen & Ming Chen

Retentive Network promotes efficient RNA language modeling of long sequences

Yi Shen[1], Guangshuo Cao[2], Yueming Hu[1], Shilong Zhang[1], Jianghong Wu[3], Dijun Chen[2, *] and Ming Chen[1, 4, *]

[1] Department of Bioinformatics, College of Life Sciences, Zhejiang University, Hangzhou 310058, China

[2] State Key Laboratory of Pharmaceutical Biotechnology, School of Life Sciences, Nanjing University, Nanjing 210023, China

[3] College of Animal Science and Technology, Inner Mongolia Minzu University, Tongliao 028000, China

[4] State Key Laboratory of Vegetation Structure, Function and Construction, College of Life Sciences, Zhejiang University, Hangzhou 310058, China

* Correspondence: dijunchen@nju.edu.cn (Dijun Chen)

* Correspondence: mchen@zju.edu.cn (Ming Chen)

Abstract

The latent features of RNA sequences are crucial for our understanding of their functions. Thus, Transformer-based nucleotide language models have received widespread attention; however, the $O(n^2)$ complexity of Transformer limits their ability to process long sequences. In this work, we propose RNAret, an RNA language model based on Retention Network, which achieves training parallelism, low computational overhead, and long-sequence processing through a retention mechanism, with $O(n)$ complexity. We pretrain RNAret using a self-supervised masked language modeling approach on 29.8 million RNA sequences. Experiments demonstrate the merit of RNAret as an RNA language model, achieving superior performance on a range of tasks, including RNA-RNA interaction prediction, RNA secondary structure prediction, and mRNA/lncRNA classification. RNAret shows strong potential for extracting latent features from RNA sequences and advancing our understanding of RNA biology.

Introduction

RNA is a fundamental component of the central dogma, playing a vital role in gene expression, protein synthesis, and various regulatory processes[1]. The ability to accurately predict RNA structure and function from its sequence holds profound biological significance and broad applicability. Unlike DNA, which is typically double-stranded, RNA is predominantly single-stranded and has various types. This diversity leads to more intricate and varied patterns and

dependencies within RNA sequences compared to DNA, making their efficient and accurate analysis a formidable challenge[2]. Traditional experimental methods for determining the features of massive RNA sequences are often costly and time-intensive, which has drawn attention to the development of machine learning methods to analyze RNA sequences.

Conventional machine learning methods for nucleotide sequences typically rely on manual feature engineering to capture essential information for specific tasks. However, this approach requires the construction of task-specific feature sets, and the feature space is difficult to generalize across different tasks[3]. Consequently, each RNA-related task requires tailored feature design, limiting the scalability and transferability of the models.

Transformer-based language models[4] have achieved remarkable success in natural language processing and autoregressive tasks[5]. Transformer architectures include encoder-only (e.g., BERT), decoder-only (e.g., GPT), and encoder-decoder (e.g., T5), each employing different attention masking mechanisms. While decoder-only models are suitable for autoregressive text generation, seq2seq models are designed for sequence-to-sequence mapping tasks, and encoder-only models are particularly suited for representation learning, thus attracting significant interest from researchers in the life sciences. Nucleotides, represented by "A, T/U, C, G", form a unique language system that encodes the genetic information of organisms. Language models are well-suited to capture the conditional distributions of patterns within nucleotide sequences, thereby modeling their intricate dependencies. RNA-FM[6], RNA-MSM[7], RNAErnie[8], and RhoFold+[9], as encoder-only RNA language models, have demonstrated advanced capabilities in various tasks. However, the $O(n^2)$ time and space complexity of the Transformer architecture poses challenges when dealing with long sequences. Some models, such as uni-RNA[10] and RiNALMo[11], employ FlashAttention[12,13] to optimize memory usage and computational pipelines, improving efficiency to some extent. Genomic models Enformer[14] and Evo[15] use convolutional layers to compress data and expand the receptive field, enabling longer context lengths. However, these approaches do not fundamentally address the computational cost of Transformer when treating long sequences.

To overcome these limitations, we propose RNAret, a pretrained RNA language model based on the Retentive Network (RetNet) architecture[16], which can be fine-tuned for various downstream tasks. RetNet employs a retention mechanism that enables parallel training, low-cost inference, and strong performance, making it particularly effective for modeling long sequences. RNAret is a lightweight and efficient model with 12 million parameters, making it accessible to academic teams equipped with consumer-grade GPUs and limited computational resources compared to commercial entities.

To assess the interpretability and biological relevance of RNAret, we evaluated its performance across multiple sub-tasks, with topics related to structure, function and type. Our analysis of both pretraining and downstream tasks demonstrates that RNAret effectively captures the features of RNA sequences. As an RNA language model, RNAret directly extracts high-dimensional features and generates task-general embeddings, thereby eliminating the need for manually designing specific features in new downstream tasks. The results confirm the feasibility of using an encoder-

only architecture based on Retentive Network for RNA language modeling and highlight its efficacy in addressing biological challenges. This work underscores the potential of leveraging advancements in large language models for biological applications, revealing the complex features embedded within RNA sequences of varying types and functions through the application of advanced algorithmic design.

Results

Self-supervised pretraining and feature extraction for RNA sequences

We developed an RNA language model based on the Retentive Network with an Encoder representation architecture, incorporating three different settings of $k \in \{1, 3, 5\}$. Here, k refers to the k-value in the K-mer tokenizer and vocabulary. During pretraining, we employed the MLM (Masked Language Model) self-supervised approach on the RNAcentral database[17], where the input consisted solely of RNA sequences without RNA type annotations (Fig. 1). For an RNA sequence of L-length, the RNAret pretraining model generates an embedding matrix of dimensions $L \times Hidden\ Dim$. The initial vocabulary sizes for $k \in \{1, 3, 5\}$ differ, resulting in significant variations in their initial loss values. As training progresses, the pretraining models with different k-values gradually converge to a loss of approximately 0.40 (Fig. 2b), which means that RNAret gradually learns statistical patterns of sequence structures.

Our analysis compares the RNA embeddings generated by the pretrained RNAret model against those from a randomly initialized model, using 5-mer statistics (feature dimension: 1024)[18] as a reference benchmark. For model embeddings, we average the RNA embedding features (feature dimension: 384). We subsample up to 10,000 for each RNA type from the RNAcentral dataset.

The pretrained 5-mer RNAret extracted embedding features cluster ncRNAs with similar types and functions in the dimensionality-reduced space, including atlases of all abundant RNA types in the pretraining dataset (Fig. 2c), 6 types of long RNA (Fig. 2d), and 5 types of small regulatory RNA (Fig. 2e). Although RNAret only learns from the masked representations of RNA sequences, without being exposed to RNA type annotations, it still successfully captures the distinctions among RNA features, with its extracted embeddings showing a more organized distribution in the low-dimensional space.

Quantitative evaluation of the t-SNE dimensionality reduction[19] includes three metrics: Silhouette Coefficient, Calinski-Harabasz Index and Davies-Bouldin Index (Supplementary Table 1). Bold metrics indicate the best performance within the group. The pretrained RNAret consistently outperforms the randomly initialized model, demonstrating excellent clustering metrics by extracting order from chaos and effectively distinguishing features of different RNA types. The 5-mer metric achieves superior performance in the Davies-Bouldin Index for long RNAs, which may be attributed to its prior extraction of RNA length information. Another pretrained RNA language model, RNA-FM has 640-dimensional embedding features and a larger number of parameters. It performs well on small RNAs but shows lower embedding

performance than RNAret on long sequences and the global RNA atlas. These results demonstrate that RNAret effectively uncovers nucleotide (and motif) correlations and conditional probabilities especially in long RNAs, successfully capturing RNA types and their functional characteristics through unsupervised learning. Therefore, RNAret is able to perform well in downstream tasks.

miRNA-mRNA interaction prediction

To assess the performance of RNAret in RNA-RNA interaction tasks, we employed the MirTarRAW dataset[20-22], which comprises 13,860 positive pairs and 13,860 negative pairs. Each pair contains a miRNA sequence and an mRNA 3'UTR sequence, as most miRNA target sites are located in the mRNA 3'UTR region. We use 72% of the dataset as the training set, 8% as the validation set, and 20% as the test set. We further use the DeepMirTarLeft dataset as an independent dataset, which comprises 443 positive pairs and 385 negative pairs.

We benchmarked our model against the baseline methods, including DeepMirTar[20] and RNA language models: RNABERT[23], RNA-MSM[7], RNA-FM[6] and RNAErnie[8], which also use embeddings of RNA sequences as features without incorporating structural or type information (RNA-MSM additionally introduces multiple sequence alignment information) (Table 1). RNAret, particularly the 5-mer model, outperformed other approaches across various metrics without feature engineering or the design of complex classifiers. In particular, 5-mer RNAret achieved an impressive F1 score of 0.962 and an accuracy of 0.962 in MirTarRAW dataset. Experiments demonstrate the strong performance of the fine-tuned 5-mer RNAret, with the [CLS] pooled features extracted by RNAret effectively distinguishing positive and negative samples in the Principal Component Analysis subspace (Fig. 3a). This observation highlights the ability of the RNAret language model in representing RNA sequences.

We calculated the average retention score map across all samples for 5-mer RNAret (Fig. 3b). The high retention scores of neighboring tokens likely reflect the need for continuous base pairing between miRNA and mRNA sequences, and vice versa. The retention scores of the [CLS] token across different positions reveal which regions contribute more significantly to the [CLS] feature representation (Fig. 3c). Higher retention scores are detected at positions 2-7, which correspond to the seed region of the miRNA, an essential segment for the binding of miRNA to mRNA[24]. After examining the dataset through browsing the starBase[25], we confirm that the peak around position 55 is close to the complementary pairing site of the seed region.

RNA secondary structure prediction

We employed two widely used benchmarks for RNA secondary structure prediction. Benchmark 1 includes the RNAStrAlign[26] and ArchiveII[27] datasets. We identified redundancy within and between both datasets, which led to data leakage. To address this issue, we removed duplicate sequences within and across the datasets, and only retained

samples with fewer than 600 nucleotides. Afterwards, we still had 20,527 samples from RNAStrAlign, which we split into training and validation sets at a 9:1 ratio, and 1,574 samples from ArchiveII, which we reserved as the test set. Benchmark 2 includes the bpRNA-1m dataset[28], which consists of three distinct subsets: TR0 (10,814 structures) for training, TV0 (1,300 structures) for validation, and TS0 (1,305 structures) for testing. By employing these deduplicated datasets, we were able to conduct a more fair and accurate comparison.

We compared RNAret and baseline models on benchmarks mentioned above. Unlike our other experimental results, in this particular task, 1-mer RNAret achieved better performance. This may be explained by the fact that RNA secondary structure prediction requires assessment of RNA base-pair interactions. Compared to base pairs (16 possible combinations, excluding [UNK]), K-mer pairs (4^k possibilities) have significantly lower abundance in the dataset, making it more challenging to train the model effectively. The 1-mer RNAret performs well across all metrics, particularly in F1 score and precision, implying that it generates fewer incorrect base pairings (Table 2).

For 1-mer RNAret, the average F1 scores vary across different RNA families in the ArchiveII dataset. Similar to other models, 1-mer RNAret has the weakest performance in the 23s rRNA family, which may be due to the absence of this family in the training set (Fig. 4a). The difficulty in achieving cross-family generalization for RNA secondary structure prediction through deep learning is widely observed[29]. Nevertheless, RNAErnie demonstrated higher F1 score compared to the widely used baseline UFold, both in the overall evaluation and across multiple RNA families (Supplementary Fig. 1). We also illustrate the relation between the F1 scores and sequence lengths, indicating that RNAret performs better on shorter sequences, a trend consistent with other models (Fig. 4b).

We display the diagonalized logits output of 1-mer RNAret for two samples (16s rRNA *H. volcanii* domain 4 and RNase P RNA *R. norvegicus*), as well as the probability maps obtained through the sigmoid activation, and the contact maps after post processing (Fig. 4c, 4e). These results demonstrate that RNAret produces robust probability maps with less noise. We make comparisons between the predictions of 1-mer RNAret, those from the UFold web server[30], and RNAfold web server[31] along with the ground truths, visualized with Forna[32] (Fig. 4d, 4f). RNAret's predictions are closer to the real structures, especially in the case of RNase P RNA *R. norvegicus*. While both UFold and RNAfold struggle with this challenging structure, RNAret still manages to reconstruct its secondary structure to a good extent.

These results show the ability of the RNAret language model in structural modeling. Additionally, the post-processing module we employed, particularly the idea of solving an assignment problem, although differs from the commonly used dynamic programming algorithms[33], has shown good feasibility.

mRNA/lncRNA classification

Predicting the coding potential of transcripts is a fundamental and crucial problem in biology. The innovative architecture of RNAret enables it to process long sequences efficiently without truncation or segmentation. Here, we utilize the

lncRNA_H and lncRNA_M datasets from RNAErnie, which are derived from protein-coding transcripts and lncRNA sequences of human and mouse in GENCODE[34]. The human-derived lncRNA_H dataset contains 77,778 training samples, 8,641 validation samples, and 21,605 test samples. The mouse-derived lncRNA_M dataset contains 37,765 training samples, 4,196 validation samples, and 10,491 test samples from mice. Notably, both datasets contain partial-length transcripts with incomplete CDS regions. As described in the method section, we evaluate the sequence-level classification performance of RNAret against the baseline models.

We perform the comparison between CPC2[35], CPAT[36] and RNA language models on the lncRNA_H and lncRNA_M datasets (Table 3). The absence of start or stop codons in partial-length transcripts limits the performance of conventional machine learning approaches that depend on identifying the longest open reading frame. RNABERT, constrained by its 440-nucleotide input limitation, fails in this context. In contrast, RNA-FM and RNA-MSM, with their 1024-nucleotide capacity, successfully capture substantial sequence patterns. RNAErnie addresses the long-sequence challenge through segmentation of sequences and aggregation of logits. Meanwhile, RNAret's architecture naturally supports long-sequence processing, enabling it to achieve superior performance, particularly on the lncRNA_H dataset, with an accuracy of 0.948.

The RNAret 5-mer fine-tuned model reveals band-like patterns in retention scores (Fig. 5a). This observation prompted us to calculate column-wise averages of retention scores for deeper analysis. We spotted that the positions with the 1% highest retention scores exhibited non-random, biologically meaningful patterns. Among these, stop codons (UAG, UAA, UGA) stood out with significantly high retention scores, despite their relatively low frequency in CDS. However, the start codon (AUG) receives only moderately high scores, likely because most AUG codons are not the actual initiation signals of CDS due to their high frequency[37]. (Fig. 5b)

We further examined the 9-mers with high retention scores and discovered a striking presence of low-complexity regions. Humans and mice exhibited similar motif distributions, with relatively lower variations in base changes within ±2 positions (Fig. 5c). Motifs such as "GCCGCCGCC", "AAAAACAAA" and "AAAUAAAAA" were particularly prominent, especially those characterized by extended poly-A stretches. These regions may play an important role in differentiating coding from non-coding regions in RNA sequences. Unbiased analysis of sequence and context preferences has proved that human RNA-binding proteins (RBPs) tend to bind low-complexity RNA motifs[38].

The computational efficiency of RNAret

To highlight the computational efficiency of RNAret, we assessed its data processing speed during the training and inference phase of our experiments. In this section, we collected the time cost of the 5-mer model. All experiments were conducted on a single A800 80GB GPU with mixed-precision autocast enabled. In the pretraining and mRNA/lncRNA classification tasks, we fully utilized the GPU memory to its maximum capacity.

During the pretraining phase, our model achieved a TPS of 10^5-level, suggesting that it can handle roughly 1.5×10^5 bases or K-mer per second under experimental conditions. In downstream tasks, computational efficiency drops due to additional classifiers. Especially in RNA secondary structure prediction, the classifier involves feature concatenation and 2D convolution, which notably slow down the speed. (Supplementary Table 2)

During inference, the absence of gradient backpropagation allows the process to run approximately twice as fast as training. However, in RNA secondary structure prediction, the additional post-processing module, which requires solving an assignment problem, becomes the primary computational bottleneck due to the $O(n^3)$ time complexity of the Hungarian algorithm (Supplementary Table 3).

We report the runtime efficiency of RNA-FM and RNA-MSM under the same environment (Supplementary Table 4). Both models exhibit lower TPS than RNAret during training and inference, and their capability to handle long sequences is also limited. In summary, RNAret manifests significant computational efficiency while maintaining good performance in biological tasks, enabling cost-effective processing of large-scale biological data.

Discussion

Retentive Network is regarded as an innovative and promising large language architecture, and we see its potential for biological applications, inspired by the idea that the retention mechanism may share similarities with the interaction between nucleotides/motifs. In this study, we develop RNAret, an innovative RNA language model built upon the RetNet. Our work introduces the successful implementation of a bidirectional Retentive Network as an Encoder representation for constructing RNA language models.

We explored different K-mer tokenization strategies and observed that larger k-values generally performed better (except in our RNA secondary structure prediction task), aligning with the findings of DNABERT[39]. This phenomenon likely stems from the ability of longer K-mers to suitably capture interactions between adjacent bases.

RNAret offers multi-purpose applications, serving as a tool for extracting RNA embeddings and a foundation for task-specific adaptation through downstream classifiers. RNAret boasts advantageous characteristics: it has a lightweight structure, ensuring efficient training and inference processes, while maintaining state-of-the-art performance across diverse downstream tasks. We employed several evaluation metrics like F1 score, Precision, Recall, accuracy and AUC, and are convinced that RNAret exhibits remarkable interpretability and robust capabilities, including in long-sequence modeling tasks.

Our research primarily concentrated on establishing the feasibility and assessing the performance of RNAret. However, we did not explore the hyperparameter configurations of RNAret, since the parameter space is too large and complex. In our code repository, we have provided training and evaluation scripts and welcome interested researchers to examine the influence of varying hyperparameters on the model's efficacy. Also, our pretraining approach includes only the MLM

task, without complex pretraining objectives based on type or structure. Moreover, RNAret is currently confined to RNA modeling and has not been expanded to the domains of DNA and protein sequences. We will further improve RNAret to make it a competitive tool for the language modeling—and potentially generation—of biomolecular sequences.

Methods

Bidirectional encoder representation from Retentive Network

The Retentive Network is an emerging language model architecture that introduces a retention mechanism in its retention layer. This layer shares structural similarities with the Transformer layer, comprising two main components: multi-scale retention (replacing multi-head self-attention) and a feedforward neural network (FFN), along with layer normalization[40] and residual connections[41]. The structure of the l^{th} retention layer can be described as follows:

$$Y_l = \mathrm{MultiScaleRetention}\big(\mathrm{LayerNorm}(X_l)\big) + X_l \tag{1}$$

$$X_{l+1} = \mathrm{FeedForwardNetwork}\big(\mathrm{LayerNorm}(Y_l)\big) + Y_l \tag{2}$$

Here, X_l is the input to the layer, and X_{l+1} is the output of the layer as well as the input to the next layer. The feedforward neural network part is computed as $\mathrm{FeedForwardNetwork}(X) = \mathrm{GELU}(XW_1)W_2$.

In the multi-scale retention mechanism, each retention head operates in parallel during both training and single-step inference. The calculation formulas are as follows:

$$Q = \big(XW_Q\big) \odot \Theta, K = (XW_K) \odot \overline{\Theta}, V = XW_V \tag{3}$$

$$\Theta_n = e^{in\theta}, D_{nm} = \begin{cases} \gamma_{n-m}, & n \geq m \\ 0, & n < m \end{cases} \tag{4}$$

$$\mathrm{Retention}(X) = (QK^{\top} \odot D)V \tag{5}$$

Here, Q and K represent the feature projections derived from the input sequence X, while D_{nm} denotes the causal masking decay matrix. The retention mechanism substantially lowers the computational cost in language models, with a $O(n)$ time and space complexity in parallel, which enables the effective modeling of long sequences. Decay factor γ and rotation matrix Θ_n are integrated into xPos rotational position encoding[42], allowing RNAret to extrapolate for longer sequences.

A multi-scale retention layer contains multiple retention heads assigned with different γ. The outputs from these heads are concatenated and subsequently projected to form the output of the multi-scale retention layer.

$$\gamma = 1 - 2^{-5-\mathrm{arange}(0,h)} \in R^h, \mathrm{head}_i = \mathrm{Retention}(X, \gamma_i) \tag{6}$$

$$Y = \mathrm{GroupNorm}_h\big(\mathrm{Concat}(\mathrm{head}_1, \cdots, \mathrm{head}_h)\big) \tag{7}$$

$$\text{MultiScaleRetention}(X) = (\text{swish}(XW_G) \odot Y)W_O \quad (8)$$

However, the original implementation of the Retentive Network is designed as a decoder-only autoregressive model, where each token can only see preceding tokens, resulting in unidirectional information retention. While this architecture is well-suited for generation tasks, it has limitations in encoder representation tasks that demand access to global context. Inspired by the idea of RMT[43], we adapt the decay matrix to

$$D_{nm} = \gamma^{|n-m|} \quad (9)$$

It implements the bidirectional retention mechanism, that is, bidirectional Retentive Network for sequence encoder representation.

RNAret model architecture

RNAret is built on the torchscale implementation and modified as described above. RNAret consists of 8 retentive layers with a feature embedding dimension of 384, an FFN embedding dimension of 512, and a value embedding dimension of 512. Other hyperparameters include: 4 retention heads, GeLU gated activation[45], 0.2 dropout rate for both the FFN and GeLU activation, and 1×10^{-6} LayerNorm epsilon value. With roughly 12 million trainable parameters, RNAret is designed as a lightweight language model.

RNA tokenization

We implemented a K-mer tokenization strategy to segment RNA sequences into continuous, overlapping tokens, following a similar approach to DNABERT[39]. Unlike simplified splitting methods, this approach captures local contextual dependencies between adjacent nucleotides. Specifically, for a given RNA sequence of length L, a sliding window of size k moves across the sequence with a stride of one nucleotide. This process generates a sequence of $L - k + 1$ K-mers from the vocabulary, which encompasses all 4^k possible nucleotide permutations.

To ensure that the final number of tokens corresponds exactly to the original sequence length L, which is crucial for base-level prediction tasks especially for the RNA secondary structure prediction, we applied a specific padding strategy using a special filler token [FIL]. We append $\frac{k-1}{2}$ [FIL] tokens to both the beginning and end of each sequence. Consequently, the input sequence is transformed into a token sequence of length L.

For instance, considering the RNA sequence 'AUGGCU' with length $L = 6$: For $k = 1$, it is tokenized directly as {A, U, G, G, C, U}, equivalent to base-level tokenization. For $k = 3$, the tokenization yields {[FIL], AUG, UGG, GGC, GCU, [FIL]}. For $k = 5$, the sequence is tokenized as {[FIL], [FIL], AUGGC, UGGCU, [FIL], [FIL]}. During the experimental phase, we trained and evaluated the model using different tokenization methods with $k \in \{1, 3, 5\}$.

In addition to 4^k K-mer tokens and [FIL] token, our vocabulary also includes several special tokens: [PAD] for aligning batch sequences to a uniform length; [UNK] for representing K-mers containing non-standard bases; [SEQ] as a separator between concatenated sequences; and [MASK], which is exclusively used during the pretraining phase to denote masked positions. The [CLS] token is introduced during fine-tuning to aggregate global sequence representations for classification tasks.

Pretraining strategies

We utilized the RNA sequences from RNAcentral release 21.0[17] as our pretraining dataset, which comprises approximately 29.8 million non-coding RNA (ncRNA) sequences from diverse families. Though the dataset excludes some RNA types such as mRNA, downstream tasks demonstrate that RNAret can generalize to RNA types not present in the pretraining dataset.

During the pretraining phase, we followed the Masked Language Model (MLM) self-supervised task, which is well-suited for encoder-only architectures. We randomly mask regions of continuous k tokens, with 15% of the tokens being masked in total (Fig. 2a). For masked tokens, 80% are replaced with [MASK] tokens, 10% are substituted with random K-mers from the vocabulary, and the remaining 10% are left unchanged[45]. The model takes the partially masked sequence as input and reconstructs the original sequences through an output projection. The optimization objective is to minimize the cross-entropy loss between the real tokens and the predicted probabilities:

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\, ln\left(\hat{y}_{ij}\right) \tag{10}$$

Each RNAret model was trained on a single A800 80GB GPU. The training configuration included a batch size of 100, a maximum sequence length of 2000, and automatic mixed precision. Training spanned approximately 2 epochs, or 600,000 steps, taking roughly 9 days to complete. By decreasing the batch size or sequence length, the training process can be replicated on consumer-grade GPUs. We employed the AdamW optimizer[46], and the initial learning rate was set at 1×10^{-4}. We employed a cosine annealing schedule to adjust the learning rate, with a cycle length of 50,000 steps and a minimum learning rate of 1×10^{-5}.

Fine-tuning strategies for downstream tasks

After the pretraining phase, RNAret has developed a foundational understanding of the distribution of bases and K-mers within RNA sequences. In this study, we evaluated RNAret on three types of downstream tasks: interaction, structure, and classification. The features used in these tasks include the pooled features extracted from the [CLS] token, the global embeddings of RNA sequences, and the fusion of embedding features. During the fine-tuning phase,

we did not freeze the parameters of RNAret.

We compared RNAret with 4 RNA language models based on Bidirectional Encoder Representations from Transformers (BERT). Their numbers of trainable parameters are as follows: RNABERT (480 kilo)[23], RNA-MSM (95.9 million)[7], RNA-FM (99.5 million)[6], and RNAErnie (105 million)[8]. For RNABERT, RNA-MSM and RNA-FM, we trained the models using the implementation of BERT-like baselines provided by RNAErnie[47], with the language model parameters unfrozen. For RNAErnie, we used the official model weights provided.

RNA-RNA interaction prediction

RNA-RNA interaction prediction focuses on determining the likelihood of interaction between two RNA sequences. The two sequences are concatenated with a separator token [SEQ], and a classification token [CLS] is prepended at the beginning of the combined sequence. The embedding features of the [CLS] token are delivered to a dense classifier for the binary interaction label.

To explain the interpretability of 5-mer RNAret, we visualized the retention scores across the test set as described: retention scores are calculated and normalized as follows:

$$R = QK^{\top} \odot D, \tilde{R}_{nm} = \frac{R_{nm}}{\max\left(\left|\sum_{i=1}^{n} R_{ni}\right|, 1\right)} \tag{11}$$

We extracted the normalized retention scores from the last retention layer and averaged them across 4 heads. To identify the most influential sequence elements that contribute to the [CLS] token representation, we specifically focus on the [CLS] row of the retention scores. Here, higher retention scores indicate that the information from the corresponding position in the sequence is more likely to be retained in the pooled features extracted from the [CLS] token.

RNA secondary structure prediction

The aim of RNA secondary structure prediction is to identify which base pairs within an RNA molecule engage in hydrogen bonding interactions, or in other words, to generate an $L \times L$ binary contact map X that satisfies the following fundamental physical constraints[48]:

(i) Base pairings are restricted to canonical pairs: G-C, A-U, and G-U;

(ii) Sharp loops are not permitted;

(iii) Duplicate pairings are not allowed. Each row and column contain no more than one "1";

(iv) The matrix must be symmetric, reflecting the bidirectional nature of base pairing interactions.

We apply outer concatenation to the RNAret embedding features for a 2D feature map, where the pairwise feature

between token s_i and s_j is represented as $[s_i\ Concat\ s_j]$ (feature dimension: 768). The feature map is passed to 16 residual blocks[42] to obtain a probability map A. Inspired by E2Efold[49], we employ a post-processing module to derive the contact map X from A.

To effectively make use of prior physical constraints, for each RNA sequence, we construct a masking matrix *M* based on rules **(i)** and **(ii)**, defined as $M_{ij} = \begin{cases} 1, & \text{if } (s_i, s_j) \text{ satisfies (i) and (ii)} \\ 0, & \text{otherwise} \end{cases}$.To satisfy rule **(iii)** and **(iv),** we set *A* as a lower triangular matrix, discarding base pairs with scores below the threshold of 0.5, which yields a filtered matrix *A′*, where $A'_{ij} = \begin{cases} A_{ij}, & \text{if } i > j \text{ and } A_{ij} >= 0.5 \\ 0, & \text{otherwise} \end{cases}$. We then frame this as an assignment problem and employ the Hungarian algorithm[50] to solve for the lower triangular matrix *X′* that maximizes the objective $\sum_{i=1}^{n}\sum_{j=1}^{n} A'_{ij} \cdot X_{ij}$, where $\sum_{j=1}^{n} X_{ij} \leq 1\ \forall i,\ \sum_{i=1}^{n} X_{ij} \leq 1\ \forall j,\ X_{ij} \in \{0,1\}\ \forall i,j$. Finally, the contact map X is computed as:

$$X_{ij} = \left(X'_{ij} \cdot M_{ij}\right) + \left(X'_{ij} \cdot M_{ij}\right)^T \quad (12)$$

Finally, any conflicting pairs introduced in this step are eliminated by a simple greedy approach.

mRNA/lncRNA classification

Current methodologies mostly attempt to distinguish mRNA from lncRNA by capturing features of whole RNA sequences, like K-mer statistics, the longest open reading frame, or pooled features[51]. By extracting these features, they determine whether a transcript is more similar to mRNA or lncRNA. In our work, we regarded this task as a base-level analysis instead of a sequence-level classification. More specifically, our strategy aims to determine the probability that each base belongs to a CDS region. For an RNA sequence of length L, the model outputs a length-L CDS probability set $\{p_1, p_2, \ldots, p_L\}$ through 16 residual blocks.

Compared to sequence-level classification methods, this approach is more fine-grained and extensible. For comparison with baseline models, we employed a sliding window strategy. A window size (WS) of 30, which is approximately the minimum length of an effective CDS, is moved along the sequence of length L and the average CDS probability $\overline{p_n} = \frac{1}{WS}\sum_{k=n}^{n+WS-1} p_k$, $1 \leq n \leq L - WS + 1$ within each window. $p = \max\{\overline{p_1}, \overline{p_2}, \ldots, \overline{p_{L-WS+1}}\}$ represents the coding potential of the sequence. Sequences with $p \geq 0.5$ are classified as mRNA, while those with $p < 0.5$ are classified as lncRNA.

We used the retention scores of the 5-mer RNAret described in the RNA-RNA Interaction Prediction section to identify the most significant promoters and motifs for CDS recognition. Specifically, for each sample in the test set, we first averaged the normalized retention scores across the 4 heads of the last retention layer and then calculated the column-wise mean. Next, we pinpointed the top 1% of high-scoring positions in each sample, and extracted the codon as well as 9-mer centered on the corresponding base. Finally, we derived comprehensive statistical insights from the lncRNA_H and lncRNA_M test sets.

Evaluation of computational efficiency

In this section, we evaluate the token processing capability of 5-mer RNAret on a single A800 GPU. Specifically, we measure the total sequence length ($Batch\ Size \times Maximum\ Length$) that the model can input and process per second while returning results.

In the pretraining phase, the model optimized one step per batch, and we recorded the time cost per step. Tokens per Second (TPS) was calculated as follows:

$$\mathrm{TPS} = \frac{\text{Maximum Sequence Length} \times \text{Batch Size}}{\text{Time per Step} \times 60} \tag{13}$$

In downstream fine-tuning phase, we measured the average time per epoch for both training and validation sets. TPS was calculated as follows:

$$\mathrm{TPS} = \frac{\text{Maximum Sequence Length} \times \text{Sample Number}}{\text{Time per Epoch} \times 60} \tag{14}$$

Statistics and reproducibility

Data processing and statistical analyses were performed on the Python platform. Specifically, we utilized biopython (1.78)[52] for processing biological data; fairscale (0.4.0), torch (2.4.0), and torchscale (0.3.0) for model construction and training; and scikit-learn (1.6.1)[53] for calculating statistical metrics (including F1 score, Accuracy, Precision, Recall, and AUC). Compatible versions of these packages are also permissible.

Sample sizes were determined based on the availability of high-quality sequences in public databases. For the ArchiveII test dataset, we excluded samples that overlapped with RNAStrAlign to prevent data leakage and ensure fair comparison. In the RNA secondary structure prediction task, sequences longer than 600 nucleotides were not considered.

To ensure reproducibility, we provide the complete code repository and pre-split datasets, facilitating the replication of the entire workflow. With the default hyperparameter settings provided in our scripts, the model training converges consistently.

Data availability

All the datasets used for analyses in this work are publicly available online. The RNAcentral dataset is available at https://ftp.ebi.ac.uk/pub/databases/RNAcentral/releases/21.0/. Datasets for downstream fine-tuning tasks are available

at https://bis.zju.edu.cn/rnaret/download/ and have been deposited in Zenodo (https://doi.org/10.5281/zenodo.18313475)[54]. Source data for figures are provided in Supplementary Data 1-3.

Code availability

The RNAret source code, including scripts for pretraining, training, and inference, is available on GitHub (https://github.com/DrBlackZJU/RNAret/) and archived on Zenodo (https://doi.org/10.5281/zenodo.18271233)[55]. The RNAret web server is accessible at https://bis.zju.edu.cn/rnaret/. Model weights are available at the project website (https://bis.zju.edu.cn/rnaret/download/) and have also been deposited on Zenodo (https://doi.org/10.5281/zenodo.18313475) [54].

References

1. Caprara, M. G. & Nilsen, T. W. RNA: versatility in form and function. *Nat. Struct. Biol*. **7**, 831-833 (2000).
2. Holbrook, S. R. RNA structure: the long and the short of it. *Curr. Opin. Struct. Biol.* **15**, 302-308 (2005).
3. Chen, Z., Ain, N. U., Zhao, Q. & Zhang, X. From tradition to innovation: conventional and deep learning frameworks in genome annotation. *Brief. Bioinform.* **25**, bbae138 (2024).
4. Vaswani, A. et al. Attention is all you need. In *Adv. Neural Inf. Process. Syst.* **30**, 5998–6008 (NIPS, 2017).
5. Min, B. et al. Recent advances in natural language processing via large pre-trained language models: a survey. *Acm Comput. Surv.* **56**, 30 (2023).
6. Chen, J. et al. Interpretable RNA foundation model from unannotated data for highly accurate RNA structure and function predictions. Preprint at https://arxiv.org/abs/2204.00300 (2022).
7. Zhang, Y. et al. Multiple sequence alignment-based RNA language model and its application to structural inference. *Nucleic Acids Res.* **52**, e3 (2024).
8. Wang, N. et al. Multi-purpose RNA language modelling with motif-aware pretraining and type-guided fine-tuning. *Nat. Mach. Intell.* **6**, 548-557 (2024).
9. Shen, T. et al. Accurate RNA 3D structure prediction using a language model-based deep learning approach. *Nat. Methods*. **21**, 2287-2298 (2024).
10. Wang, X. et al. Uni-RNA: universal pre-trained models revolutionize RNA research. Preprint at https://www.biorxiv.org/content/10.1101/2023.07.11.548588v1 (2023).
11. Penić, R. J., Vlašić, T., Huber, R. G., Wan, Y. & Šikić, M. RiNALMo: general-purpose RNA language models can generalize well on structure prediction tasks. *Nat. Commun.* **16**, 5671 (2025).
12. Dao, T., Fu, D., Ermon, S., Rudra, A. & Ré, C. FlashAttention: fast and memory-efficient exact attention with IO-Awareness. In *Adv. Neural Inf. Process. Syst.* **35,** 16344–16359 (NIPS, 2022).
13. Dao, T. FlashAttention-2: faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations* (ICLR, 2024)
14. Avsec, Ž. et al. Effective gene expression prediction from sequence by integrating long-range interactions. *Nat. Methods*. **18**, 1196-1203 (2021).
15. Nguyen, E. et al. Sequence modeling and design from molecular to genome scale with Evo. *Science*. **386**, eado9336 (2024).
16. Sun, Y. et al. Retentive Network: a successor to Transformer for large language models. Preprint at https://arxiv.org/abs/2307.08621 (2023).
17. Sweeney, B. et al. RNAcentral 2021: secondary structure integration, improved sequence search and new member databases. *Nucleic Acids Res.* **49**, D212-D220 (2021).
18. Kirk, J. M. et al. Functional classification of long non-coding RNAs by k-mer content. *Nat. Genet.* **50**, 1474-1482 (2018).
19. van der Maaten, L. & Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579-2605 (2008).
20. Wen, M., Cong, P., Zhang, Z., Lu, H. & Li, T. DeepMirTar: a deep-learning approach for predicting human miRNA targets. *Bioinformatics*. **34**, 3781-3787 (2018).
21. Pla, A., Zhong, X. & Rayner, S. miRAW: A deep learning-based approach to predict microRNA targets by analyzing whole microRNA transcripts. *PLoS Comput. Biol.* **14**, e1006185 (2018).

22. Gu, T., Zhao, X., Barbazuk, W. B. & Lee, J. miTAR: a hybrid deep learning-based approach for predicting miRNA targets. *BMC Bioinform*. **22**, 96 (2021).
23. Akiyama, M. & Sakakibara, Y. Informative RNA base embedding for RNA structural alignment and clustering by deep representation learning. *NAR Genom. Bioinform.* **4**, lqac12 (2022).
24. Bartel, D. P. MicroRNAs: target recognition and regulatory functions. *Cell*. **136**, 215-233 (2009).
25. Li, J., Liu, S., Zhou, H., Qu, L. & Yang, J. starBase v2.0: decoding miRNA-ceRNA, miRNA-ncRNA and protein-RNA interaction networks from large-scale CLIP-Seq data. *Nucleic Acids Res.* **42**, D92-D97 (2014).
26. Tan, Z., Fu, Y., Sharma, G. & Mathews, D. H. TurboFold II: RNA structural alignment and secondary structure prediction informed by multiple homologs. *Nucleic Acids Res.* **45**, 11570-11581 (2017).
27. Sloma, M. F. & Mathews, D. H. Exact calculation of loop formation probability identifies folding motifs in RNA secondary structures. *RNA*. **22**, 1808-1818 (2016).
28. Danaee, P. et al. bpRNA: large-scale automated annotation and analysis of RNA secondary structure. *Nucleic Acids Res.* **46**, 5381-5394 (2018).
29. Szikszai, M., Wise, M., Datta, A., Ward, M. & Mathews, D. H. Deep learning models for RNA secondary structure prediction (probably) do not generalize across families. *Bioinformatics*. **38**, 3892-3899 (2022).
30. Fu, L. et al. UFold: fast and accurate RNA secondary structure prediction with deep learning. *Nucleic Acids Res.* **50**, e14 (2022).
31. Gruber, A. R., Lorenz, R., Bernhart, S. H., Neuböck, R. & Hofacker, I. L. The Vienna RNA websuite. *Nucleic Acids Res.* **36**, W70-W74 (2008).
32. Kerpedjiev, P., Hammer, S. & Hofacker, I. L. Forna (force-directed RNA): Simple and effective online RNA secondary structure diagrams. *Bioinformatics*. **31**, 3377-3379 (2015).
33. Zuker, M. & Stiegler, P. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.* **9**, 133-148 (1981).
34. Frankish, A. et al. GENCODE 2021. *Nucleic Acids Res.* **49**, D916-D923 (2021).
35. Kang, Y. et al. CPC2: a fast and accurate coding potential calculator based on sequence intrinsic features. *Nucleic Acids Res.* **45**, W12-W16 (2017).
36. Wang, L. et al. CPAT: Coding-Potential Assessment Tool using an alignment-free logistic regression model. *Nucleic Acids Res.* **41**, e74 (2013).
37. Subramanian, K., Payne, B., Feyertag, F. & Alvarez-Ponce, D. The codon statistics database: a database of codon usage bias. *Mol. Biol. Evol.* **39**, msac157 (2022).
38. Dominguez, D. et al. Sequence, structure, and context preferences of human RNA binding proteins. *Mol. Cell*. **70**, 854-867 (2018).
39. Ji, Y., Zhou, Z., Liu, H. & Davuluri, R. V. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics*. **37**, 2112-2120 (2021).
40. Ba, J. L., Kiros, J. R. & Hinton, G. E. Layer normalization. Preprint at https://arxiv.org/abs/1607.06450 (2016).
41. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 770–778 (IEEE, 2016).
42. Sun, Y. et al. A length-extrapolatable Transformer. In *61st Annual Meeting of the Association-for-Computational-Linguistics* 14590-14604 (ACL, 2023)
43. Fan, Q., Huang, H., Chen, M., Liu, H. & He, R. RMT: Retentive Networks meet Vision Transformers. In *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR, 2024).

44. Hendrycks, D. & Gimpel, K. Gaussian error linear units (GELUs). Preprint at https://arxiv.org/abs/1606.08415 (2016).
45. Kenton, J. & Toutanova, L. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL-HLT 2019* Vol. 1 (eds Burstein, J. et al.) 4171–4186 (Association for Computational Linguistics, 2019).
46. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations* (ICLR, 2019).
47. Ning, W. CatIIIIIIII/RNAErnie: v.1.0. Zenodo https://doi.org/10.5281/zenodo.10847621 (2024).
48. Nowakowski, J. & Tinoco, I. RNA structure and stability. *Semin. Virol*. **8**, 153-165 (1997).
49. Chen, X., Li, Y., Umarov, R., Gao, X. & Song, L. RNA secondary structure prediction by learning unrolled algorithms. In *International Conference on Learning Representations* (ICLR, 2020).
50. Kuhn, H. The Hungarian Method for the assignment problem. *Nav. Res. Logist.* **52**, 7-21 (2005).
51. Ventola, G. M. M. et al. Identification of long non-coding transcripts with feature selection: a comparative study. *BMC Bioinform*. **18**, 187 (2017).
52. Cock, P. J. A. *et al.* Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25**, 1422–1423 (2009).
53. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
54. Shen, Y. RNAret - Datasets and Model Weights [Data set]. *Zenodo* https://doi.org/10.5281/zenodo.18313475 (2026).
55. Shen, Y. DrBlackZJU/RNAret: Retentive Network promotes efficient RNA language modeling of long sequences (v1.0). *Zenodo* https://doi.org/10.5281/zenodo.18271233 (2026).

Acknowledgements

This work was partially supported by the National Key Research and Development Program of China [2023YFE0112300]; National Science Foundation of China [32261133526; 32570787]; the Science and Technology Innovation Leading Scientist [2022R52035], the 151 talent project of Zhejiang Province (first level); and Collaborative Innovation Center for Modern Crop Production co-sponsored by province and ministry. The authors are grateful to the members of Ming Chen's laboratory for helpful discussions and valuable comments, and Jianghong Wu for assistance with computational resources.

Author contributions

M.C. and D.C. supervised and designed the study. Y.S. designed the study, implemented the model and performed data analysis with support from J.W. and Y.H. Y.S. wrote the manuscript with input from G.C. and Y.H. S.Z. helped to build up the web server. All authors reviewed and approved the submitted manuscript.

Additional information

Competing interests: The authors declare no competing interests.

Tables

Table 1: Performance of RNAret on miRNA–mRNA interaction prediction task

	MirTarRAW				
	F1	**Precision**	**Recall**	**Accuracy**	**AUC**
DeepMirTar	0.9235	0.9479	0.9235	0.9348	0.9793
RNA-FM	0.9489	0.9490	0.9489	0.9490	0.9763
RNABERT	0.8965	0.8978	0.8963	0.8966	0.9484
RNA-MSM	0.9529	0.9529	0.9530	0.9529	0.9894
RNAErnie	0.9568	0.9569	0.9569	0.9568	0.9886
1-mer RNAret	0.9471	0.9360	0.9585	0.9461	0.9856
3-mer RNAret	0.9568	0.9433	**0.9706**	0.9558	0.9899
5-mer RNAret	**0.9622**	**0.9627**	0.9617	**0.9619**	**0.9911**
	DeepMirTarLeft				
	F1	**Precision**	**Recall**	**Accuracy**	**AUC**
DeepMirTar	-	-	-	-	-
RNA-FM	0.9648	0.9644	0.9654	0.9650	0.9847
RNABERT	0.8367	0.8368	0.8384	0.8370	0.8933
RNA-MSM	0.9672	0.9673	0.9671	0.9674	0.9870
RNAErnie	0.9634	0.9631	0.9637	0.9635	0.9920
1-mer RNAret	0.9583	0.9572	0.9594	0.9553	0.9889
3-mer RNAret	0.9599	0.9473	**0.9729**	0.9565	0.9903
5-mer RNAret	**0.9728**	**0.9772**	0.9684	**0.9710**	**0.9949**

Bold formatting indicates the best results on the metrics.

Table 2: Performance of RNAret on RNA secondary structure prediction task

	ArchiveII			bpRNA TS0		
	F1	**Precision**	**Recall**	**F1**	**Precision**	**Recall**
Ufold	0.8372	0.8139	0.8676	0.6301	0.5823	**0.7185**
RNA-FM	0.8188	0.8337	0.8113	0.5874	0.5582	0.6512
RNABERT	0.6971	0.7213	0.6996	0.5139	0.5174	0.5459
RNA-MSM	0.6934	0.7222	0.6750	0.5305	0.5168	0.5815
RNAErnie	0.8653	0.8680	0.8668	0.6084	0.5757	0.6721
1-mer RNAret	**0.8898**	0.9170	**0.8735**	**0.6537**	**0.6766**	0.6606
3-mer RNAret	**0.8898**	**0.9185**	0.8718	0.6410	0.6765	0.6395
5-mer RNAret	0.8728	0.9046	0.8535	0.6126	0.6506	0.6108

Bold formatting indicates the best results on the metrics.

Table 3: Performance of RNAret on mRNA/lncRNA classification task

	lncRNA_H				
	F1	**Precision**	**Recall**	**Accuracy**	**AUC**
CPC2	0.8074	0.5983	0.9490	0.7737	0.8794
CPAT	0.8696	0.7520	0.9600	0.8560	0.9418
RNA-FM	0.9218	0.9218	0.9219	0.9218	0.9691
RNABERT	0.7264	0.7335	0.7283	0.7277	0.7650
RNA-MSM	0.9076	0.9094	0.9079	0.9077	0.9601
RNAErnie	0.9419	0.9420	0.9420	0.9406	0.9833
1-mer RNAret	0.9261	0.9123	0.9404	0.9245	0.9749
3-mer RNAret	0.9335	0.9090	**0.9593**	0.9311	0.9809
5-mer RNAret	**0.9480**	**0.9504**	0.9456	**0.9477**	**0.9849**
	lncRNA_M				
	F1	**Precision**	**Recall**	**Accuracy**	**AUC**
CPC2	0.8368	0.6493	0.9717	0.8105	0.9061
CPAT	0.8939	0.7753	0.9897	0.8825	0.9648
RNA-FM	0.8966	0.8981	0.8967	0.8967	0.9564
RNABERT	0.6666	0.6701	0.6677	0.6677	0.6478
RNA-MSM	0.8853	0.8854	0.8853	0.8853	0.9487
RNAErnie	**0.9313**	0.9321	**0.9306**	**0.9340**	**0.9795**
1-mer RNAret	0.9103	0.9312	0.8904	0.9123	0.9676
3-mer RNAret	0.9288	**0.9457**	0.9125	0.9300	0.9749
5-mer RNAret	0.9295	0.9401	0.9192	0.9303	0.9765

Bold formatting indicates the best results on the metrics.

Figure Legend

Fig. 1: Overview of the design of RNAret model and its pretraining and application pipelines

RNAret consists of 8 RetNet layers and contains approximately 12 million trainable parameters. During the pretraining process, RNAret performs a masked language modeling (MLM) task on the RNAcentral dataset, which includes 29.8 million RNA sequences, attempting to reconstruct the masked K-mers. In the fine-tuning process, RNAret is paired with downstream classifiers to complete various tasks, including miRNA-mRNA interaction prediction, RNA secondary structure prediction, and lncRNA/mRNA classification.

Fig. 2: RNAret captures RNA features and patterns after pretraining

a, Tokenization and masking strategy of RNAret under different k values ($k \in \{1, 3, 5\}$). When $k > 1$, continuous blocks of k tokens are masked (grey boxes) to prevent information leakage from the middle nucleotide. **b**, The Masked Language Modeling (MLM) cross-entropy loss curve for models with different k values over 600,000 pretraining steps. The curves correspond to the 1-mer (purple line), 3-mer (green line), and 5-mer (tangerine line) models. The orange horizontal line represents 0.4, which is approximately the convergence value of the loss. **c**, t-SNE dimensionality reduction of embeddings for different RNA types in the RNAcentral dataset, including pretrained RNAret (left), RNA-FM (middle), and 5-mer feature vectors (right). Each dot represents a randomly subsampled sequence, colored according to its RNA type as indicated in the legend below, up to 10,000 samples per RNA type**. d**, t-SNE projections of several long RNA types, including rRNA, RNase P RNA, telomerase RNA, tmRNA, lncRNA and RNase MRP RNA. **e**, t-SNE projections of several short regulatory RNA types, including piRNA, snRNA, siRNA, miRNA and snoRNA.

Fig. 3: Interpretability of RNAret in miRNA-mRNA interactions prediction

a, Visualization of dimensionally reduced [CLS] token features of test samples ($n = 5{,}544$) embedded from 5-mer RNAret. This includes fine-tuned RNAret, RNAret that is only pretrained without fine-tuning, and models with random initialization. Plots show the separation of positive interaction pairs (orange dots) and negative interaction pairs (blue dots). **b**, Heatmap of the average normalized retention scores across all samples. Red regions indicate high retention scores, while blue regions indicate low scores. The black dashed lines mark the boundary of miRNA and mRNA 3' UTR. **c**, The average retention score of all samples at the [CLS] token. Red bars represent positive retention scores, and blue bars represent negative retention scores. The vertical grey dashed line separates the miRNA from the mRNA 3'UTR. High retention score regions are associated with miRNA seed regions and the complementary pairing regions of mRNA 3' UTR.

Fig. 4: Detailed performance evaluation of RNAret in secondary structure prediction on the ArchiveII dataset

a, Violin plots showing the distribution of F1 scores for the 1-mer RNAret model across different RNA families in the ArchiveII dataset ($n = 1{,}574$). The test sizes for each family are: 5s ($n = 81$), 16s ($n = 484$), 23s ($n = 21$), tRNA ($n = 276$), tmRNA ($n = 3$), grp1 ($n = 83$), srp ($n = 216$), and RNaseP ($n = 410$). Each black dot represents an individual sample. All telomerase RNA samples have appeared in the RNAStrAlign training set, so this RNA family is excluded. **b**, Scatter plot correlating F1 scores (blue dots) with sequence length in the ArchiveII dataset. The orange line represents the linear trend, indicating performance relative to sequence length. Similar to other models, RNAret performs better on shorter samples. **c, e**, Visualization of the model outputs for two representative RNA sequences: *H. volcanii* 16s rRNA domain 4 (128 nt) (**c**) and *R. norvegicus* RNase P RNA (257 nt) (**e**). Panels show, from left to right: the raw logits heatmap, the predicted probability matrix, the binary contact map derived after post-processing, and the ground truth contact map. The probability map and contact map constructed from RNAret output are closely aligned with the ground truth. **d, f**, Comparison of predicted secondary structures corresponding to the sequences in **c** and **e**. Structures are visualized using Forna, where nucleotides are colored by structural element (green for stems, blue for hairpin loops, and red for multiloops).

Fig. 5: Analysis of high retention score motifs and codons in mRNA/lncRNA classification

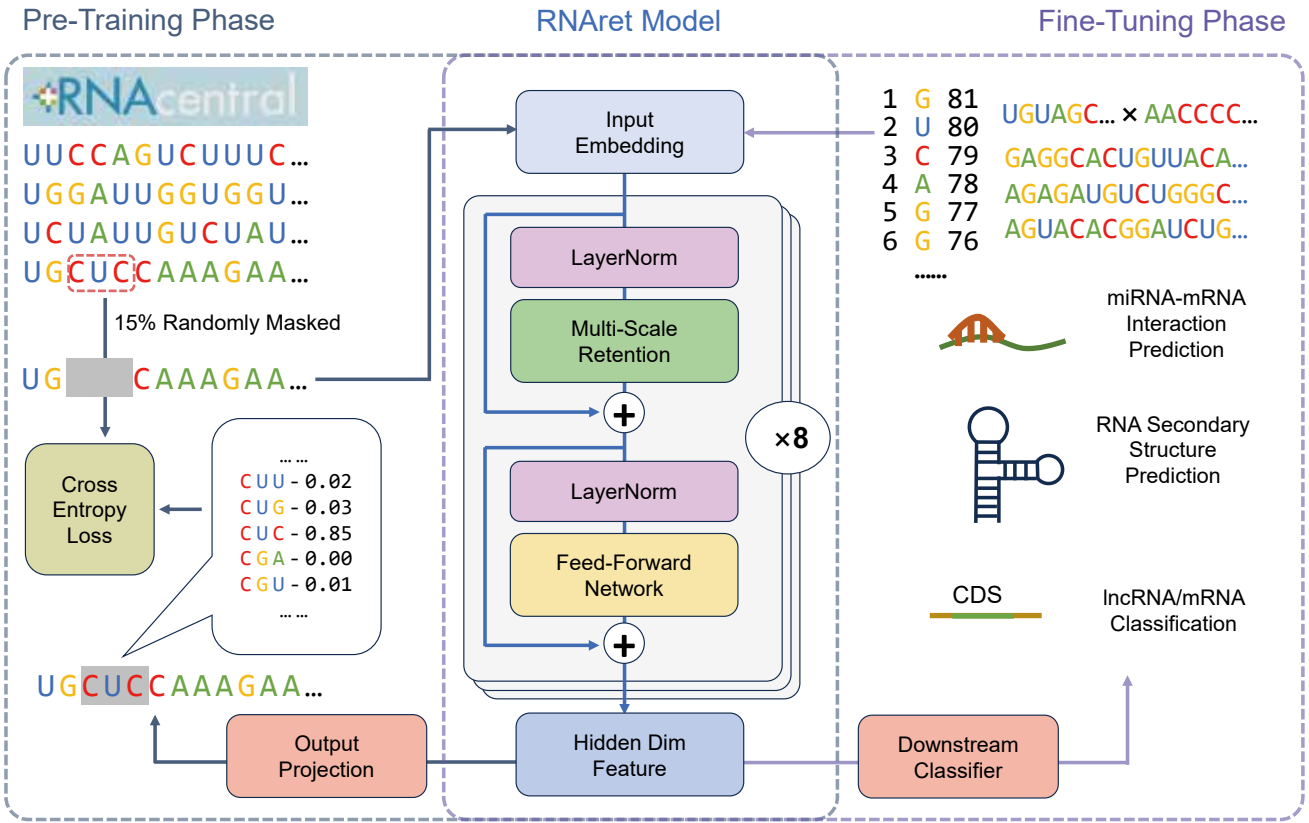
a, Visualization of normalized retention scores for a representative mRNA sample (ENST00000403162.7, left) and a lncRNA sample (ENST00000441152.3, right) in the lncRNA_H test set ($n = 21{,}605$). The heatmap displays the $L * L$ retention score map, while the line graph illustrates the average retention score for each column of the heatmap. **b**, Frequency distribution of codons corresponding to the top 1% positions with the highest retention scores in the lncRNA_H ($n = 21{,}605$) and lncRNA_M ($n = 10{,}491$) test sets. Light blue bars represent human data, and orange bars represent mouse data. The vertical green dashed line marks the theoretical frequency of a uniform random distribution (1/64). **c**, The motif distribution in the top 1% highest scoring positions for Human (top) and Mouse (bottom). The height of the nucleotide letters (A, C, G, U) corresponds to the information content (bits) at each position. Sequence logos reveal a similar pattern between human and mouse.

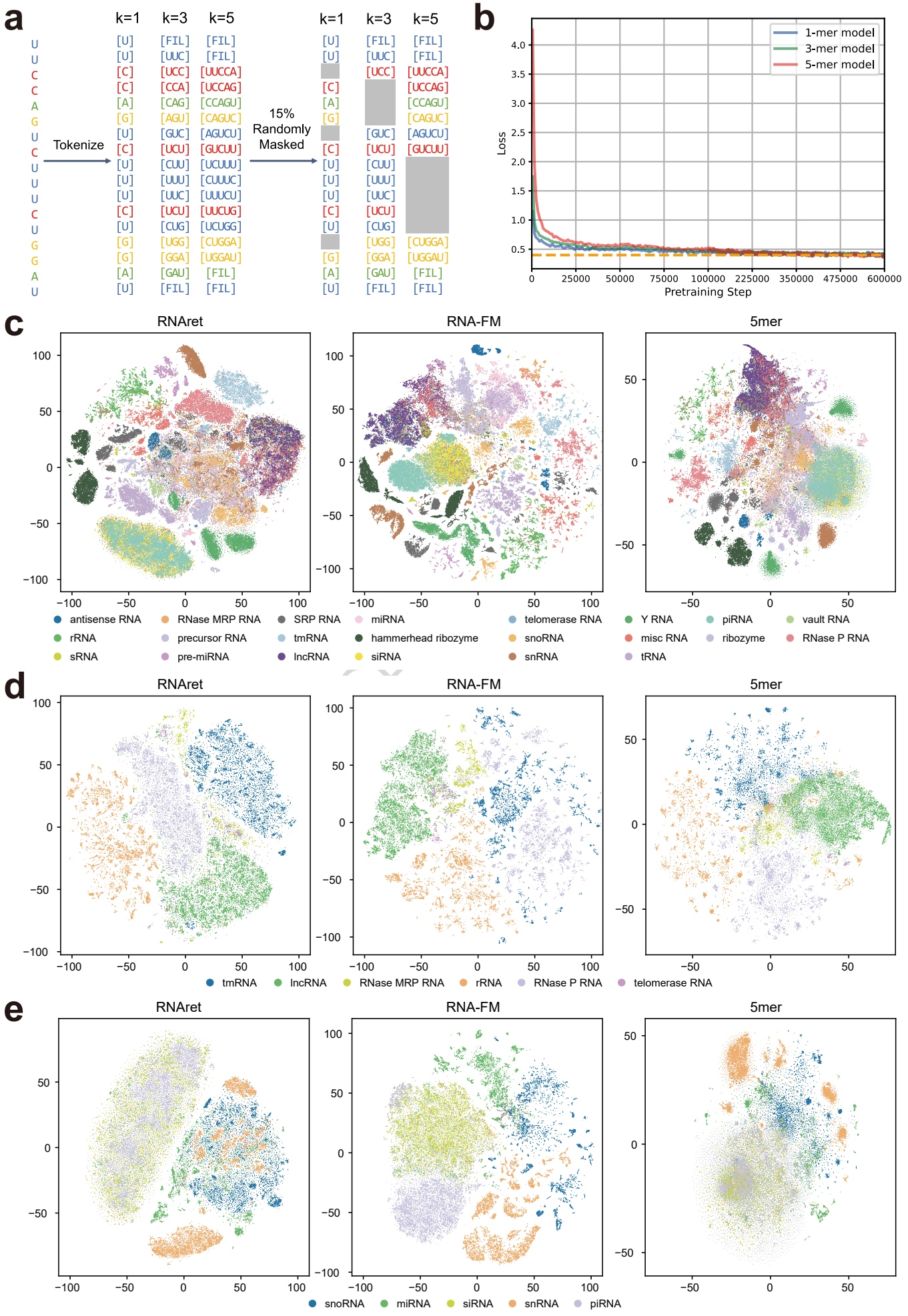
Editor's Summary

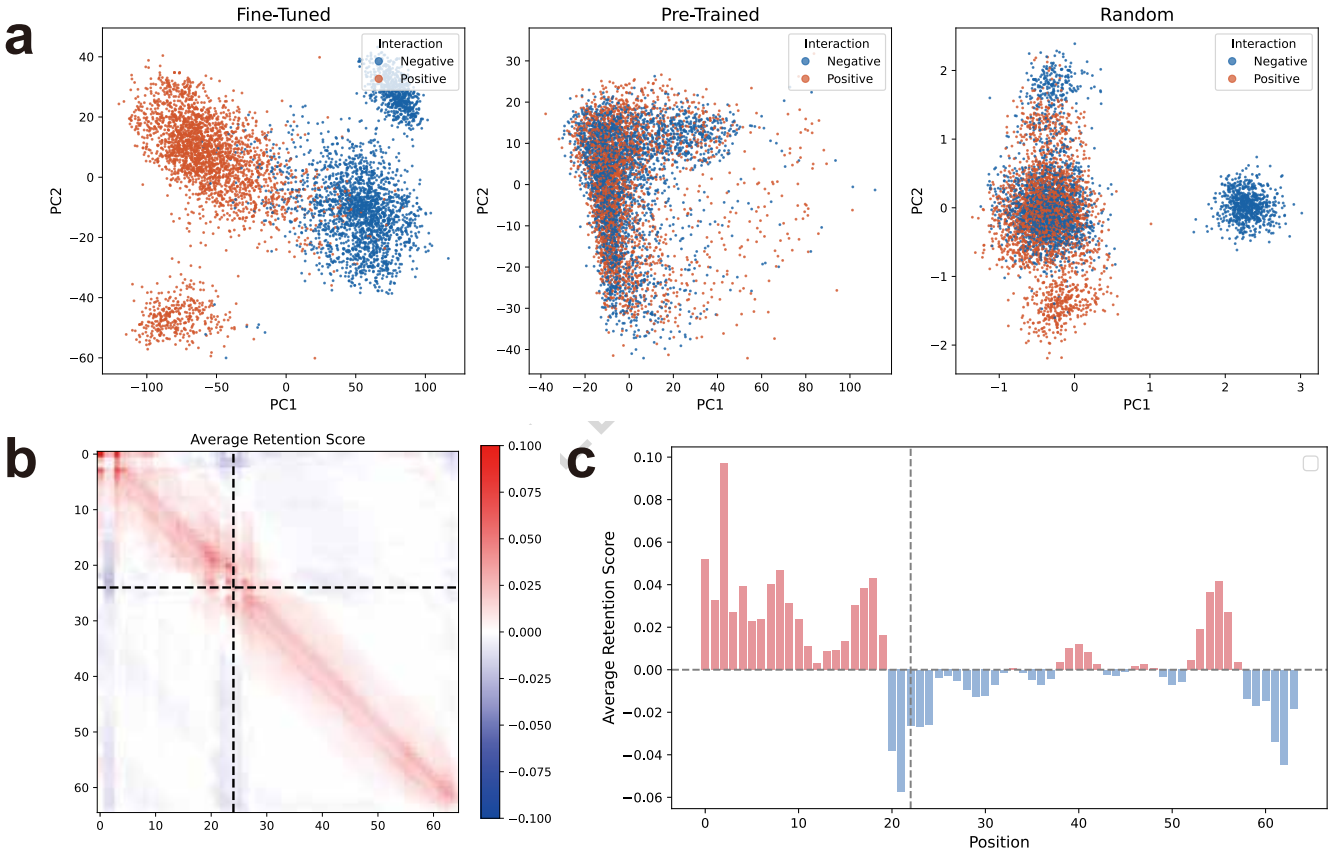
RNAret introduces a Retentive Network-based architecture to RNA modeling, enabling linear complexity for long sequences and demonstrating superior performance in predicting RNA interactions, secondary structures, and coding potential.

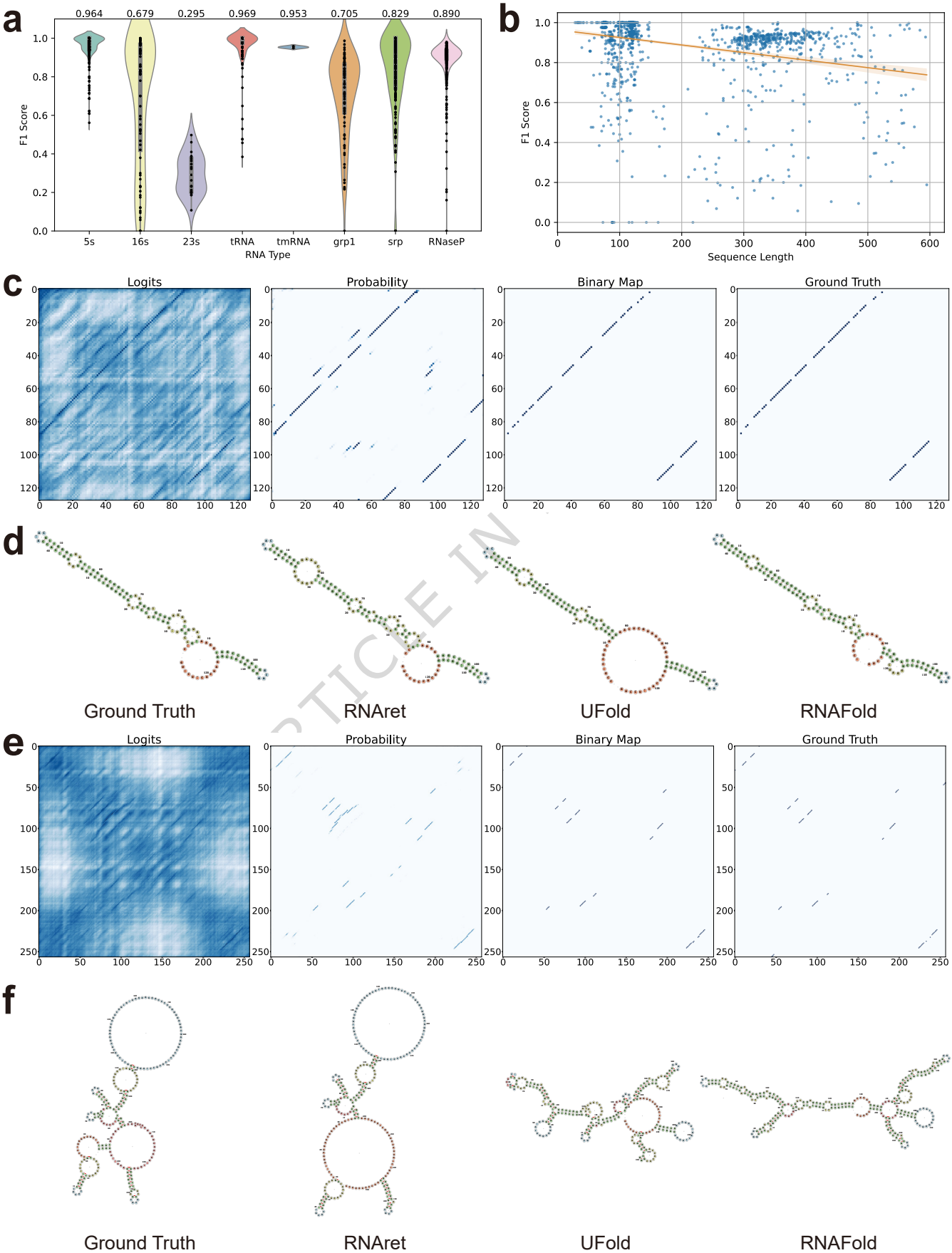
Peer review information

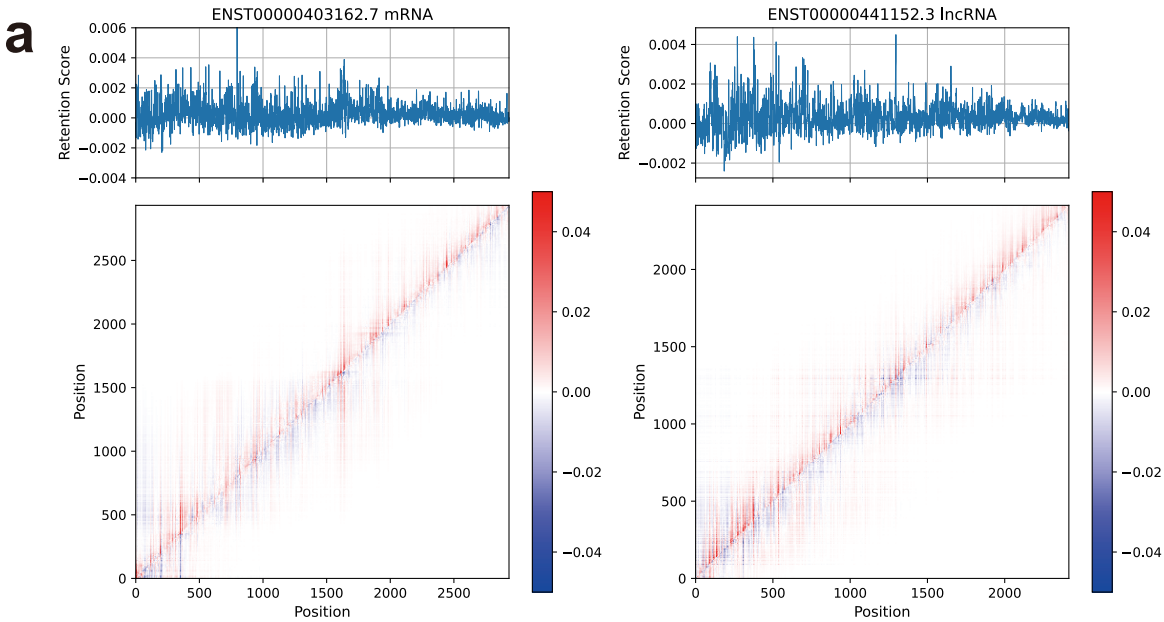
Communications Biology thanks the anonymous reviewers for their contribution to the peer review of this work. Primary Handling Editors: Professor Maria Anisimova and Dr. Nilanjan Banerjee, Dr Aylin Bircan, Dr Kaliya Georgieva. A peer review file is available.

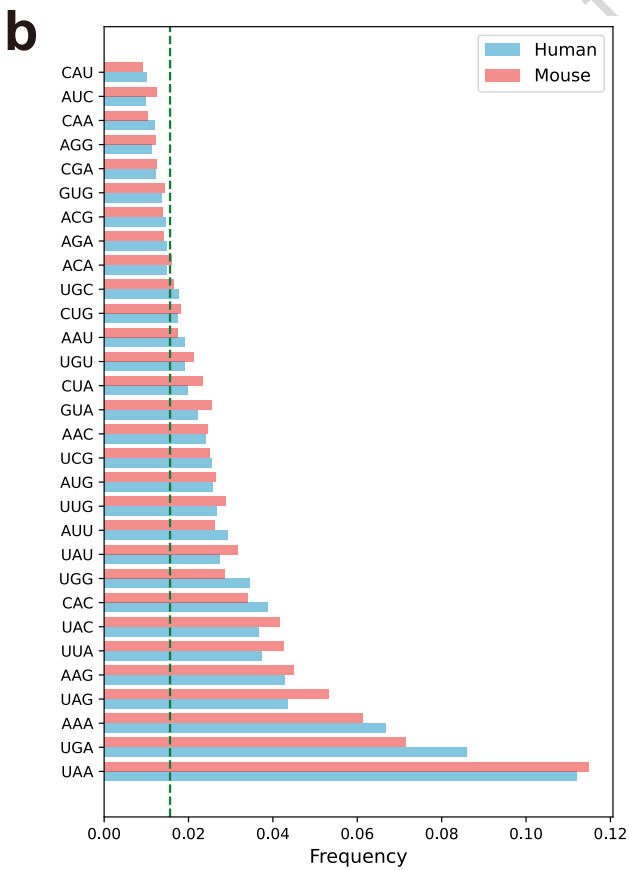

Pre-Training Phase
RNAret Model
Fine-Tuning Phase
RNAcentral
UUCCAGUCUUUC…
UGGAUUGGUGGU…
UCUAUUGUCUAU…
UGCUCCAAAGAA…
15% Randomly Masked
UG CAAAGAA…
Cross Entropy Loss
CUU - 0.02
CUG - 0.03
CUC - 0.85
CGA - 0.00
CGU - 0.01
UGCUCCAAAGAA…
Output Projection
Input Embedding
LayerNorm
Multi-Scale Retention
LayerNorm
Feed-Forward Network
×8
Hidden Dim Feature
1 G 81
2 U 80
3 C 79
4 A 78
5 G 77
6 G 76
UGUAGC… × AACCCC…
GAGGCACUGUUACA…
AGAGAUGUCUGGGC…
AGUACACGGAUCUG…
Downstream Classifier
miRNA-mRNA Interaction Prediction
RNA Secondary Structure Prediction
CDS
lncRNA/mRNA Classification

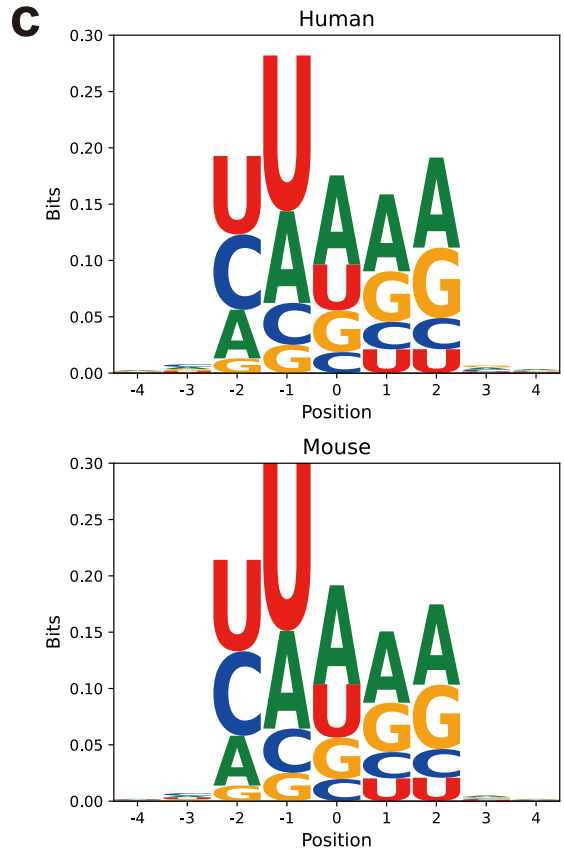

a
k=1
k=3
k=5
Tokenize
15%
Randomly
Masked
b
1-mer model
3-mer model
5-mer model
Loss
Pretraining Step
c
RNAret
RNA-FM
5mer
antisense RNA
rRNA
sRNA
RNase MRP RNA
precursor RNA
pre-miRNA
SRP RNA
tmRNA
lncRNA
miRNA
hammerhead ribozyme
siRNA
telomerase RNA
snoRNA
snRNA
Y RNA
misc RNA
tRNA
piRNA
ribozyme
vault RNA
RNase P RNA
d
RNAret
RNA-FM
5mer
tmRNA
lncRNA
RNase MRP RNA
rRNA
RNase P RNA
telomerase RNA
e
RNAret
RNA-FM
5mer
snoRNA
miRNA
siRNA
snRNA
piRNA


a
Fine-Tuned
Pre-Trained
Random
Interaction
Negative
Positive
PC1
PC2
b
Average Retention Score
c
Average Retention Score
Position


a
0.964
0.679
0.295
0.969
0.953
0.705
0.829
0.890
F1 Score
RNA Type
5s
16s
23s
tRNA
tmRNA
grp1
srp
RNaseP
b
F1 Score
Sequence Length
c
Logits
Probability
Binary Map
Ground Truth
d
Ground Truth
RNAret
UFold
RNAFold
e
Logits
Probability
Binary Map
Ground Truth
f
Ground Truth
RNAret
UFold
RNAFold


a
ENST00000403162.7 mRNA
ENST00000441152.3 lncRNA
Retention Score
Position


b
Human
Mouse
CAU
AUC
CAA
AGG
CGA
GUG
ACG
AGA
ACA
UGC
CUG
AAU
UGU
CUA
GUA
AAC
UCG
AUG
UUG
AUU
UAU
UGG
CAC
UAC
UUA
AAG
UAG
AAA
UGA
UAA
Frequency


c
Human
Mouse
Bits
Position