nature communications

Article

https://doi.org/10.1038/s41467-025-60872-5

RiNALMo: general-purpose RNA language models can generalize well on structure prediction tasks

Received: 21 October 2024

Accepted: 6 June 2025

Published online: 01 July 2025

Check for updates

Rafael Josip Penić[1], Tin Vlašić[2], Roland G. Huber[3], Yue Wan[2] & Mile Šikić[1,2]

While RNA has recently been recognized as an interesting small-molecule drug target, many challenges remain to be addressed before we take full advantage of it. This emphasizes the necessity to improve our understanding of its structures and functions. Over the years, sequencing technologies have produced an enormous amount of unlabeled RNA data, which hides a huge potential. Motivated by the successes of protein language models, we introduce RiboNucleic Acid Language Model (RiNALMo) to unveil the hidden code of RNA. RiNALMo is the largest RNA language model to date, with 650M parameters pre-trained on 36M non-coding RNA sequences from several databases. It can extract hidden knowledge and capture the underlying structure information implicitly embedded within the RNA sequences. RiNALMo achieves state-of-the-art results on several downstream tasks. Notably, we show that its generalization capabilities overcome the inability of other deep learning methods for secondary structure prediction to generalize on unseen RNA families.

Large language models (LLMs) trained on massive text corpora have been performing remarkably on various natural language understanding and generation tasks[1–6]. In recent years, the exploration of language models (LMs) has gone beyond the domain of natural language processing (NLP), reaching into the realms of biology and its data. A vast amount of sequenced protein data provided a ground for training protein LMs, and since they have proven to be an extremely valuable asset in protein generative[7–9] and structure prediction tasks[10,11].

Most efforts in applying the ideas originally developed for NLP have been focused on proteins after the success of AlphaFold[12] in the prediction of protein structures. ESM-1b[13] was one of the first models that applied the self-supervised language modeling approaches to protein data. It was pre-trained on 250M protein sequences and tested on several downstream tasks, including the secondary structure and tertiary contact prediction, where it achieved state-of-the-art results. Later on, several other protein LMs were proposed and tested on various downstream tasks[14–17]. Protein LMs play an important role in protein tertiary structure prediction. ESM-2[11] and OmegaPLM[10] are examples of protein LMs that efficiently replace a multiple sequence alignment (MSA) step in deep learning (DL) methods for structure prediction.

RNAs play crucial roles in fundamental biological processes, including transcription, cell signaling, chromatin remodeling, and genome imprinting. Like proteins, RNAs have recently become an attractive drug target, whose function and interaction with other molecules are closely related to their structure[18,19]. However, much less attention has been given to applying LMs to RNA-related problems, partly because there is no such amount of available data and corresponding structures, and partly because similar problems tend to be more difficult than for proteins. Furthermore, RNA structure prediction is severely hindered by the scarcity of high-resolution structural

[1]Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia. [2]Genome Institute of Singapore (GIS), Agency for Science Technology and Research (A*STAR), Singapore, Republic of Singapore. [3]Bioinformatics Institute (BII), Agency for Science, Technology and Research (A*STAR), Singapore, Republic of Singapore. e-mail: mile_sikic@gis.a-star.edu.sg

data and the lack of unbiased, diverse sequence alignments, which are often used as a source of evolutionary information[20].

Currently, there are only two single-input-sequence RNA foundation models, RNA-FM[21] and Uni-RNA[22], that have found applications in several structure and function prediction tasks. RNA-FM is a 100M parameters Transformer encoder based on the original implementation by ref. 23 and trained exclusively on 23.7M non-coding RNAs (ncRNAs) from the RNAcentral database[24,22] pre-trained an ensemble of LMs ranging from 25M to 400M parameters trained on a much larger dataset of 1B sequences with the architecture analogous to the ESM protein LM[13] enhanced by several advanced techniques such as RoPE and fused layer norm. The authors pre-trained language models of different sizes and reported when the model parameters exceeded 400M, the performance in downstream tasks reached a plateau with their architectures and datasets. Both foundation models used the standard BERT-style masked language modeling (MLM) pre-training task[1]. Unlike the Uni-RNA, RNA-FM is publicly available.

Besides the RNA-FM and Uni-RNA foundation models, several authors proposed LMs to solve a few specific downstream tasks. RNA-MSM[25], an MSA-based BERT-style RNA LM, specialized in particular for secondary structure prediction, that instead of a single input sequence utilizes a set of homologous sequences. However, obtaining the MSAs is a very time-consuming procedure—it takes RNAcmap, a homology search tool used by RNA-MSM, on average 9 h to obtain an MSA for one RNA sequence of length 60[25,26] proposed SpliceBERT, a BERT-style encoder pre-trained exclusively on more than 2M precursor messenger RNA (pre-mRNA) sequences from different vertebrates for studying RNA splicing. SpliceBERT outperforms DNABERT[27], an LM trained only on a human genome, both on human and non-human splice-site prediction tasks. It demonstrates better generalization capability of LMs pre-trained on multiple species[28]. Proposed single-cell BERT (scBERT), a BERT-style LM pre-trained on huge amounts of unlabeled single-cell RNA-seq data for cell type annotation. BigRNA[29] is an LM pre-trained on the genomes of 70 individuals on a task to predict DNA-matched RNA-seq data. BigRNA accurately predicts tissue-specific RNA expression and the binding sites of proteins and microRNAs. UTR-LM[30] is an RNA language model for 5′ untranslated region (5′ UTR) of mRNAs, which was pre-trained on endogenous 5′ UTRs from multiple species. It is specialized for mRNA translation-related downstream tasks such as mRNA translation efficiency and expression level prediction.

Motivated by the recent successes of protein LMs and the latest architectural improvements in LLMs, we propose RiNALMo, a novel RNA language model. We pre-trained RiNALMo on a set of carefully curated 36M ncRNA sequences from the RNAcentral database augmented by several other RNA databases. RiNALMo is a 650M parameters BERT-style Transformer encoder advanced by modern architectural techniques such as rotary positional embedding (RoPE)[31], SwiGLU activation function[32], and FlashAttention-2[33]. During pre-training, RiNALMo can extract hidden knowledge and capture the underlying structural information embedded within the sequences at the single-nucleotide level. Later, its output embeddings serve as a powerful sequence representation that improves the performance on various structural and functional RNA downstream tasks compared to other foundation models and state-of-the-art methods. In particular, RiNALMo shows remarkable generalization capability on secondary structure prediction of RNA families not encountered in the training dataset where other DL methods fail.

The main contributions of the paper are as follows:

- We propose RiNALMo, a 650M parameters RNA LM, which is the largest RNA language model to date that can fully leverage the potential of a vast amount of public unannotated RNA sequences;
- We show that the generalization capability of RiNALMo can overcome the problem of other DL methods for secondary structure prediction to perform well on RNA families not seen in the training dataset;
- We conducted extensive experiments on several RNA structural and functional downstream tasks whose results show that RiNALMo outperforms other RNA LMs and DL methods on most datasets.
- We release the pre-trained and fine-tuned RiNALMo weights and scripts for fine-tuning the model for the downstream tasks.

Results

General-purpose RNA language model

A schematic diagram of RiNALMo and its pre-training procedure and downstream tasks is shown in Fig. 1. Our LM is a Transformer encoder focused on understanding and unveiling the RNA code. At the heart of the model is the self-attention mechanism[23], which captures important local and global contextual information. We pre-trained RiNALMo using the MLM, where we tasked the model to reconstruct corrupted, unlabeled RNA sequences. In this paper, each nucleotide is a single token. To corrupt the input sequence, we randomly mask 15% of the tokens in the training sequence. To reconstruct the masked tokens, RiNALMo's embeddings are utilized by the MLM prediction head whose outputs are used in the cross-entropy loss function. More technical details, pretraining details, and ablation study are given in "Methods" and Supplementary Information.

Once pre-trained, RiNALMo's output embeddings can serve as a powerful sequence representation that has embedded structural and evolutionary information. First, its embeddings can be used for visualization and clustering analysis of RNA sequences. Second, such a representation can be used as an enriched input to structural and functional downstream tasks. We employed RiNALMo in a few tasks to assess its performance and generalization capabilities. Namely, we show how RiNALMo can improve and generalize well on secondary structure, multi-species splice-site, translation efficiency (TE), expression level (EL), and mean ribosome loading (MRL) prediction tasks as well as for multi-class ncRNA family classification tasks. However, we anticipate it can be leveraged in many other tasks related to RNA structure and function. Particularly interesting would be the employment of RiNALMo in RNA tertiary structure prediction tasks, where, motivated by the results from ESMFold[11] and OmegaFold[10], we believe RiNALMo's embeddings can successfully replace the MSA.

To analyze the interpretability of our model, we visualized pre-trained RiNALMo's sequence representations by applying t-SNE on the classification token embeddings for RNAs from a secondary structure prediction dataset and an ncRNA functional family classification dataset (see Fig. 2). RNA structures and functions vary across different RNA families, and we expect RiNALMo has learned these properties during the MLM pre-training and is able to encode them within its RNA sequence representations. We compared the classification token embeddings of RiNALMo and RNA-FM[21]. Contrary to the RNA-FM's embedding space, in the RiNALMo's embedding space, the RNAs are clustered by families with, in general, clean boundaries between clusters. This is especially evident when comparing Fig. 2c and Fig. 2d, which illustrate the embeddings of RNAs from the ncRNA functional family classification dataset. The analyses of embeddings revealed that RNAs with similar structure and function properties are grouped, implicating that RiNALMo has learned these properties beyond their primary structure. Furthermore, the t-SNE visualization shows RiNALMo's ability to cluster and distinguish different RNA families and confirms it can be used in various clustering analyses of RNA sequences. Later in "Results", RiNALMo's ability to reason beyond RNA primary structure was additionally backed by the state-of-the-art performance on an ncRNA Rfam family classification downstream task. The t-SNE visualization is important from the RNA structure perspective as well, since the RNAs from the same families fold similarly, meaning

https://doi.org/10.1038/s41467-025-60872-5

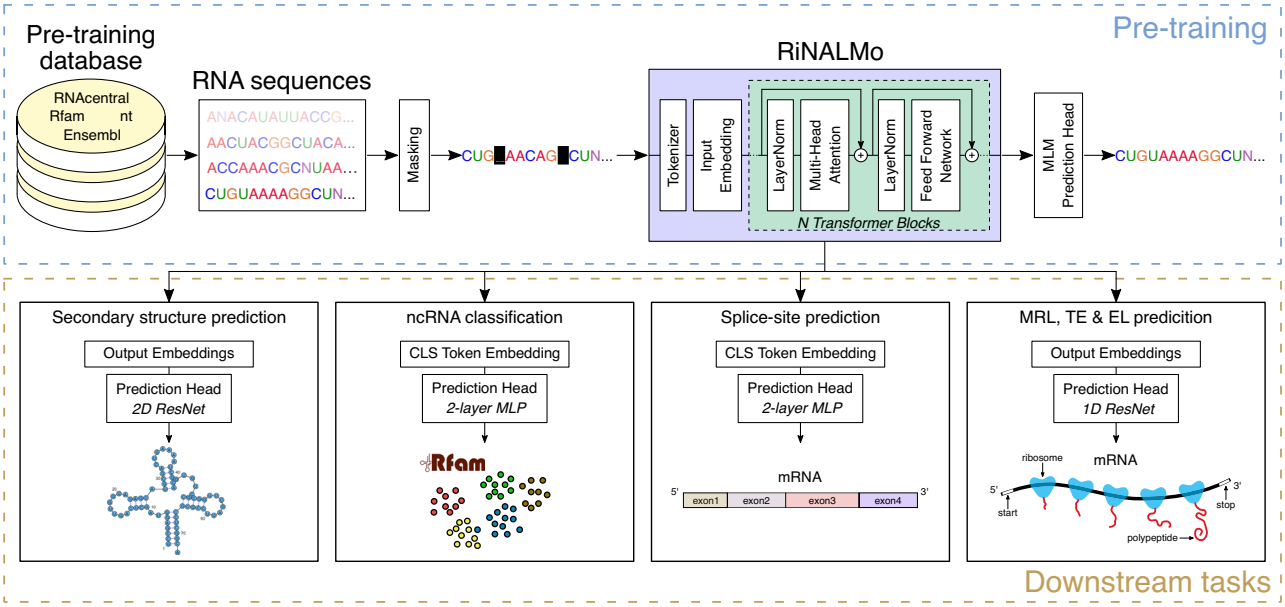


Fig. 1 | RiNALMo pre-training and applications. In the pre-training stage, RiNALMo is trained on unlabeled RNA sequences from several databases using masked language modeling (MLM). To corrupt the input sequence, we randomly mask 15% of the tokens in the training sequence. Before being passed to the Transformer, an RNA sequence is tokenized and turned into a 1280-dimensional vector using a learned input embedding module. The language model comprises 33 Transformer blocks. Each Transformer block consists of a multi-head attention and a feed-forward network. Once pre-trained, RiNALMo can be separately fine-tuned for various structural and functional downstream tasks in which its expressive output embeddings, utilized by the prediction heads, significantly improve performance. In this work, we fine-tuned RiNALMo for secondary structure, multi-species splice-site, mean ribosome loading (MRL), translation efficiency (TE), and expression level (EL) prediction, as well as for ncRNA family classification.

the structure information is also contained in the embedding space. This was later backed up by the RiNALMo's state-of-the-art performance on the inter-family generalization secondary structure prediction downstream task.

Fine-tuning RiNALMo for intra-family secondary structure prediction

When RNAs fold into complex structures, many of their bases pair up and form hydrogen bonds. These pairs are vital for the structure's stability and function. These bonds can be represented by secondary structure, which can tell us a lot about RNA and which is often used as an input to the tertiary structure prediction tools. An example of a secondary structure can be seen in Fig. 3a.

Most popular secondary structure prediction tools often rely on thermodynamic models, aiming to identify secondary structures that possess the lowest free energy[34]. There are also popular probabilistic methods based on statistical learning procedures that act as an alternative to free energy minimization methods, such as CONTRAfold[35]. Several DL methods have been developed as well. They often outperform the thermodynamic models on RNA families on which they were trained, i.e., on in-distribution data.

We fine-tuned RiNALMo on a simple binary classification task with binary cross-entropy loss, where we tasked the model to classify each nucleotide pair as either paired or unpaired. The pipeline for determining secondary structures is illustrated in Fig. 3b. We utilized a dataset proposed in ref. 36 and compared our model to RNA-FM[21] and popular DL methods specialized for secondary structure prediction SPOT-RNA[36], UFold[37], and MXfold2[38]. The proposed dataset is derived from the bpRNA database[39] which compiled secondary structures from seven different sources. Most structures were obtained with comparative sequence analysis while a smaller portion was extracted from atomic coordinates from PDB[40] using the annotation tool RNAView[41]. Authors filtered out redundant RNAs by clustering similar sequences which yielded 13,419 non-redundant secondary structures which were then randomly split into training, validation and testing datasets denoted as TR0, VL0 and TS0, respectively. All models were trained on the same training dataset (TR0) except SPOT-RNA which was additionally fine-tuned on a smaller dataset derived from PDB[40]. As can be seen in Fig. 3c, RiNALMo outperforms other state-of-the-art DL approaches in terms of precision, recall and consequently F1 score. We provide F1 score distributions in Fig. 3d. A TS0 target example and the predictions from different DL methods are given in Fig. 3g.

Beyond sequence similarity, it is important to consider structure similarity and evaluate the ability of structure prediction tools to generalize to RNAs structurally dissimilar to ones found in the training datasets. Therefore, RiNALMo was further evaluated using the datasets TrainSetA and TestSetB proposed by Rivas et al.[42]. TrainSetA consists of RNAs collected from datasets proposed in several different studies[35,43,44]. To ensure sequence diversity, similar sequences were removed, leaving a total of 3166 RNAs in the dataset. TestSetB contains RNAs from 22 Rfam[45] families with known structures not found in TrainSetA, making it structurally dissimilar. RNAs from these families are then filtered by removing similar sequences, resulting in 430 RNAs in the TestSetB. The model was first fine-tuned with RNAs from the TrainSetA dataset. All RNAs in this dataset longer than 500 nucleotides were used for validation. Once trained, the model was evaluated on the TestSetB. RiNALMo's performance on the TestSetB dataset was compared to RNAstructure[34], ContraFold[46], MXfold2[38], and RNA-FM[21]. As shown in Fig. 3e, f, RiNALMo outperforms other tools in terms of F1 score.

RNA language models can generalize well on inter-family structure prediction tasks

While DL methods for secondary structure prediction outperform thermodynamic models on in-distribution data, they are usually unable to generalize well on new RNA families[47,48]. This is a severe limitation as it hinders the practical usage of such tools.

To test the generalization capabilities of RiNALMo, we utilized the benchmark proposed by ref. 47. The benchmark utilizes the ArchiveII

https://doi.org/10.1038/s41467-025-60872-5

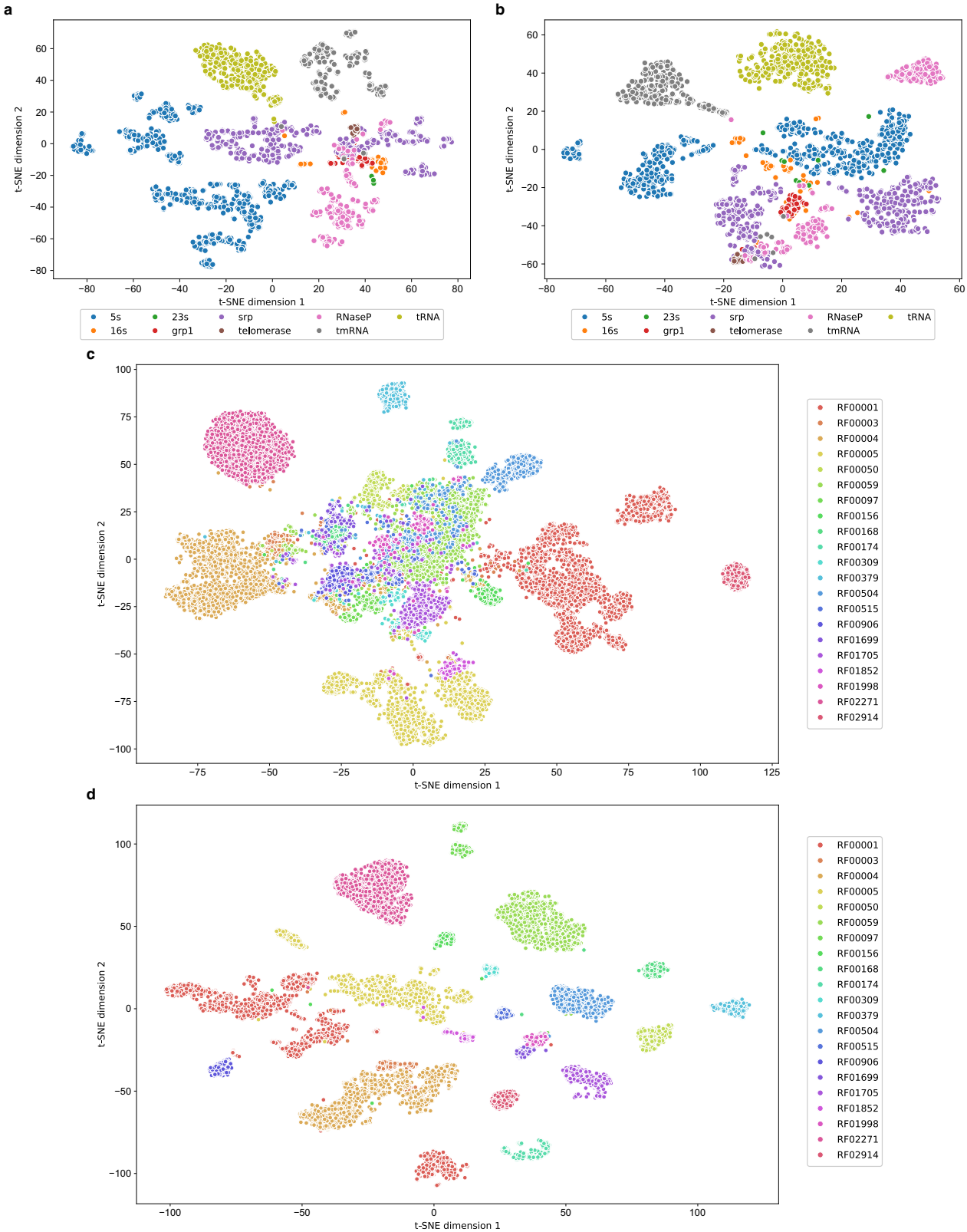
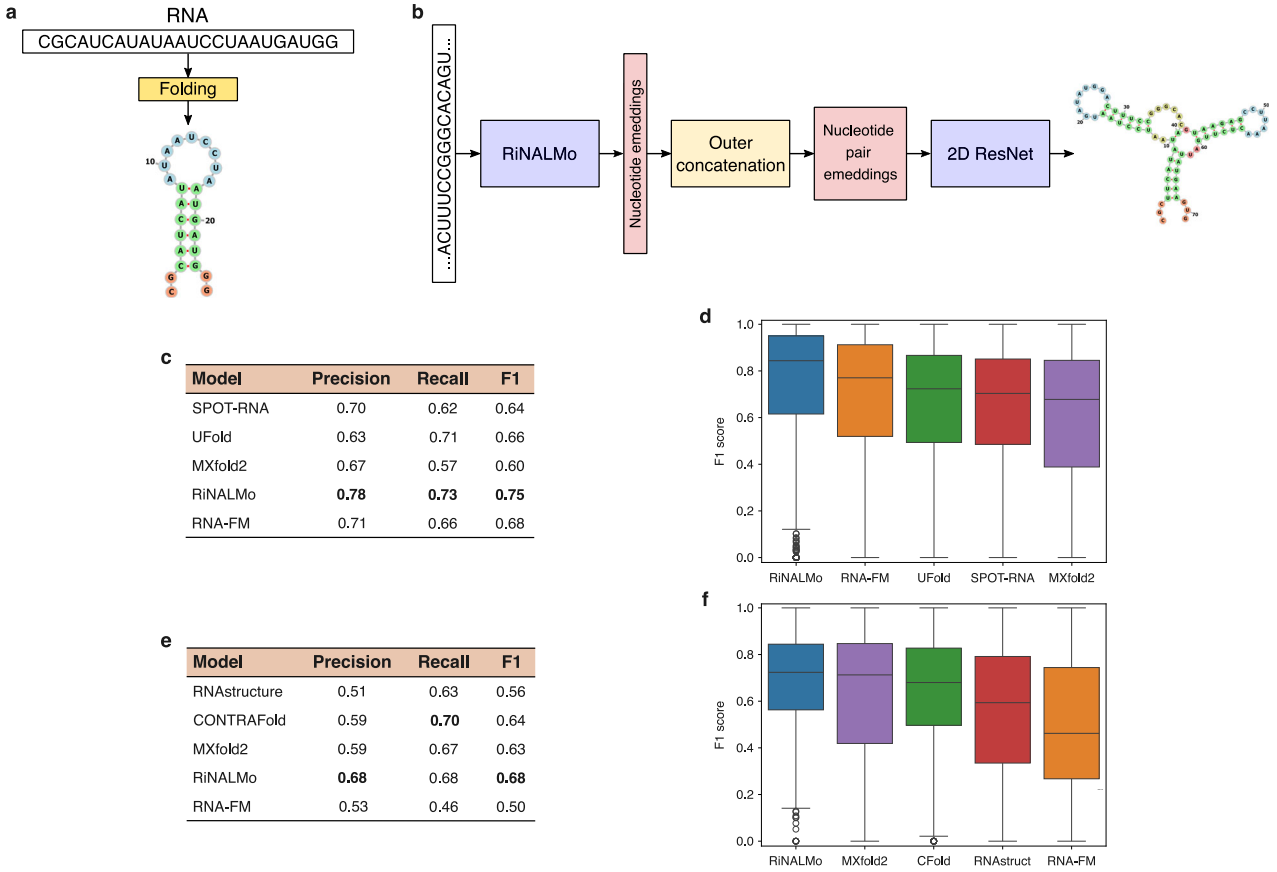


Fig. 2 | t-SNE visualizations of RNA sequence embeddings outputted by RNA-FM[21] and RiNALMo. RNA-FM (**a**) and RiNALMo (**b**) classification token embeddings for the inter-family generalization evaluation dataset. RNA-FM (**c**) and RiNALMo (**d**) classification token embeddings for one part of the ncRNA functional family classification task dataset.

dataset[49,50] of 3865 RNAs from nine families which was split nine times, and in each split, a different family was held out for evaluation while the other eight families were used for training and validation. RiNALMo's ability to generalize across different RNA families was compared to RNA-FM[21], popular thermodynamics-based tool RNAstructure[34], widely used probabilistic method CONTRAfold[35], and two DL approaches specialized for secondary structure prediction UFold[37] and MXFold2[38]. The LMs were fine-tuned and the other two DL models

https://doi.org/10.1038/s41467-025-60872-5



c

Model	Precision	Recall	F1
SPOT-RNA	0.70	0.62	0.64
UFold	0.63	0.71	0.66
MXfold2	0.67	0.57	0.60
RiNALMo	**0.78**	**0.73**	**0.75**
RNA-FM	0.71	0.66	0.68

e

Model	Precision	Recall	F1
RNAstructure	0.51	0.63	0.56
CONTRAFold	0.59	**0.70**	0.64
MXfold2	0.59	0.67	0.63
RiNALMo	**0.68**	0.68	**0.68**
RNA-FM	0.53	0.46	0.50

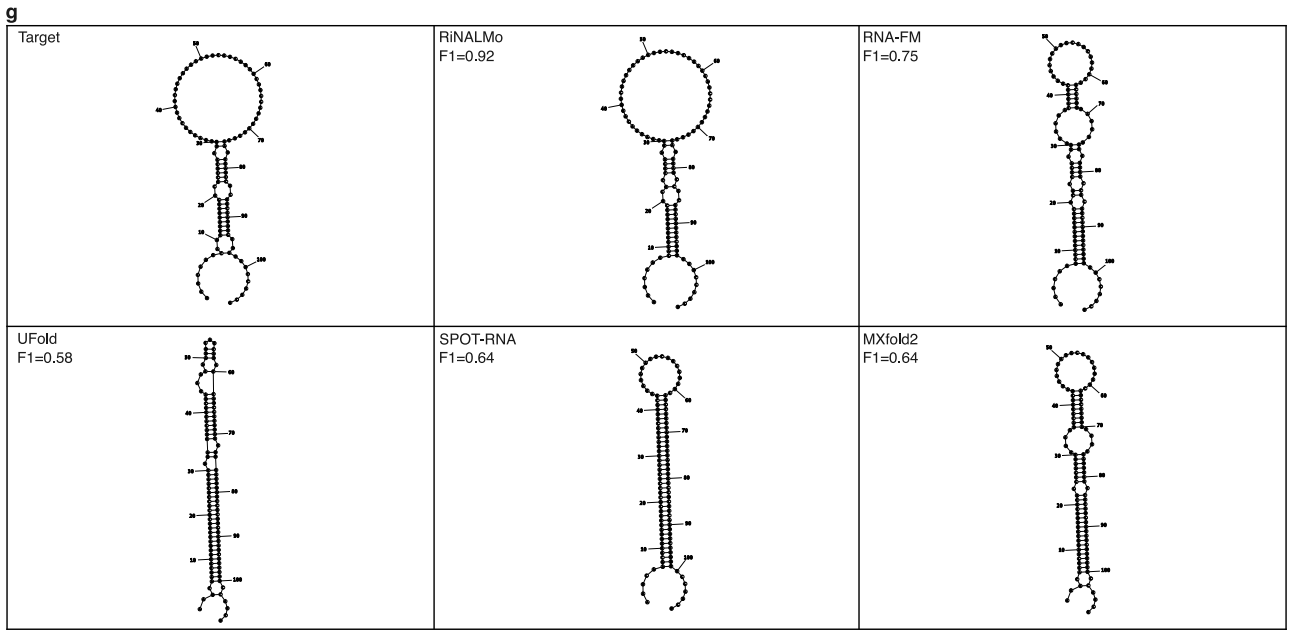


Fig. 3 | Secondary structure prediction. a RNAs fold into various shapes according to their function and while doing so, many of their nucleotides pair up using a hydrogen bond. These pairings are crucial for structural stability and form structural motifs such as hairpin loops and bulges. **b**, RiNALMo produces nucleotide embeddings for the given RNA sequence. Nucleotide pair embeddings are constructed by applying outer concatenation to RiNALMo's outputs. Finally, pair representations are fed into the convolutional bottleneck residual neural network (ResNet) which produces base pairing probabilities that are then converted into the final secondary structure prediction. **c**, Precision, recall and F1 performance of different deep learning models on the TS0 evaluation dataset. **d** Distribution of F1 scores for predictions of different models on the TS0 dataset (sample size $n = 1305$). **e** Precision, recall and F1 performance of different structure prediction tools on the TestSetB evaluation dataset. **f** Distribution of F1 scores for predictions of different structure prediction tools on the TestSetB dataset ($n = 430$). Cfold denotes CONTRAFold and RNAstruct denotes RNAstructure. **g** A target RNA from the TS0 evaluation dataset and its predictions from different deep learning models. In (**c**, **e**), the best result for each metric is shown in bold. In (**d**, **f**), Box plots show the median (center line), 25th and 75th percentiles (bounds of box), whiskers extending to the smallest and largest values within 1.5× the interquartile range, and individual outliers beyond the whiskers.

https://doi.org/10.1038/s41467-025-60872-5

a

Test Family	RNAstructure	CONTRAfold	RiNALMo	RNA-FM	MXfold2	UFold
5S rRNA	0.63	0.67	**0.88**	0.52	0.54	0.53
SRP RNA	0.63	0.60	**0.70**	0.25	0.50	0.26
tRNA	0.70	0.76	**0.93**	0.78	0.64	0.26
tmRNA	0.43	0.44	**0.80**	0.29	0.46	0.40
RNase P RNA	0.55	0.60	**0.80**	0.30	0.51	0.41
Group I intron	0.54	0.59	**0.66**	0.16	0.45	0.45
16S rRNA	0.57	0.60	**0.74**	0.13	0.55	0.41
Telomerase RNA	0.50	0.54	0.12	0.08	0.34	**0.80**
23S rRNA	0.73	0.75	**0.85**	0.17	0.64	0.45
Mean	0.59	0.62	**0.72**	0.30	0.51	0.44

b

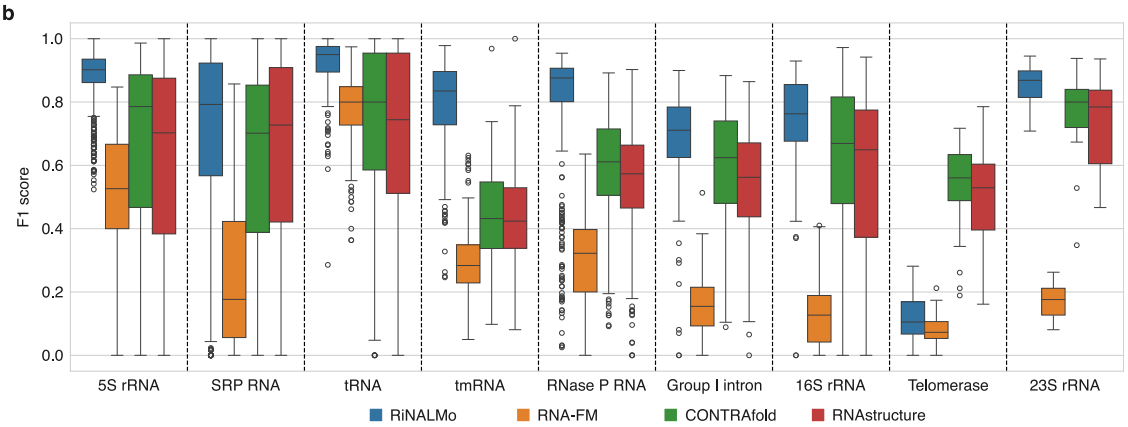


Fig. 4 | Inter-family secondary structure prediction. a, Average F1 scores for secondary structure prediction on the ArchiveII evaluation datasets. The best result for each evaluation dataset in the tables is shown in bold. **b**, Distribution of F1 scores for different methods on the ArchiveII evaluation datasets (sample sizes *n*: 1, 283, 918, 557, 462, 454, 74, 67, 35, and 15, respectively). Box plots show the median (center line), 25th and 75th percentiles (bounds of box), whiskers extending to the smallest and largest values within 1.5× the interquartile range, and individual outliers beyond the whiskers. Minimum and maximum values are 0 and 1, respectively.

were separately trained on each of the previously described dataset splits and evaluated on a corresponding unseen RNA family. For predicting the secondary structures using CONTRAfold, we used EternaFold parameters[46] trained on the EternaBench dataset, a set of more than 20,000 RNAs.

Average F1 scores and F1 score distributions for different dataset splits are shown in Fig 4. Fine-tuned RiNALMo demonstrates that it is capable of inter-family generalization as it outperforms RNAstructure and CONTRAfold in eight out of nine families by high margins, unlike other DL models. To the best of our knowledge, this is the first paper to show that LMs can generalize well on inter-family secondary structure prediction, mitigating the limitations of other DL methods. We noted, however, that RiNALMo struggles to generalize on telomerase RNAs, but it achieves the highest F1 score on all other families. Visualization of RiNALMo's sequence embeddings for all RNAs in the dataset is presented in Fig. 2b. One can notice that telomerase RNAs are clustered together, however, there is no clear boundary between them and SRP RNAs. We also noticed that telomerase RNAs are the longest in the dataset, on average around 25% longer than the second-longest in the dataset. Please refer to Supplementary Table S2 for the mean lengths and standard deviations of the RNA families in the ArchiveII dataset. Interestingly, UFold performs best on telomerase RNA, while achieving much worse results on the other families. We are currently unable to conclude why RiNALMo fails on telomerase RNAs, but we will take more focus on this problem in the future. Technical details and more secondary structure prediction results and examples can be found in the Secondary Structure Prediction section, Supplementary Note 2 and Supplementary Fig. S3.

RiNALMo's secondary structure prediction generalization capabilities were also compared to homology-based tools CentroidHomfold[51], locARNA[52], and CentroidAlifold[53]. To identify RNA homologs, we used the approach presented in ref. 54. Using the Infernal's *cmscan*[55] tool, we first determine which RNA family the target RNA belongs to. Then, from the seed alignment of the identified family, we randomly select up to 19 RNAs that share between 65% and 95% sequence identity with the target sequence. The identified homologs and the target RNA sequence are forwarded to CentroidHomfold and locARNA. Besides the structure prediction, locARNA also outputs alignment of given RNAs which is then forwarded to CentroidAlifold since it requires sequence alignment as an input. It is worth noting that locARNA and CentroidAlifold output consensus structure prediction, so to obtain the structure prediction of the target RNA, the consensus structure is mapped onto the target sequence. To ensure a fair comparison, RNAs for which no homologs were found have been excluded from the evaluation. 16S and 23S rRNAs were not included in the comparison as they are split into independent folding domains, making the identification of homologs more challenging than for other families. Except for telomerase RNAs, RiNALMo outperformed or showed comparable performance to other tools across all families. A detailed comparison of RiNALMo's performance with homology-based methods can be found in Fig. 5.

Fine-tuning RiNALMo for classification tasks

RiNALMo can also be fine-tuned for downstream tasks that determine important functions of the observed RNA. Splice-site prediction and determination of the ncRNA family are two important functional tasks that can be cast as classification tasks.

RNA splicing plays an important role in eukaryotic gene expression, involving the removal of introns from pre-mRNAs and the ligation

https://doi.org/10.1038/s41467-025-60872-5

a

Test Family	RiNALMo	CentroidHomfold	locARNA	CentroidAlifold
5S rRNA (1278 / 1283)	**0.88**	0.70	0.87	0.86
SRP RNA (738 / 918)	0.70	0.54	**0.77**	0.75
tRNA (510 / 557)	0.94	0.89	**0.97**	0.96
tmRNA (456 / 462)	**0.81**	0.24	0.60	0.53
RNase P RNA (442 / 454)	**0.81**	0.51	0.64	0.60
Group I intron (37 / 74)	**0.74**	0.60	0.37	0.43
Telomerase (35 / 35)	0.12	0.57	**0.63**	0.50
Mean	**0.71**	0.58	0.69	0.66

b

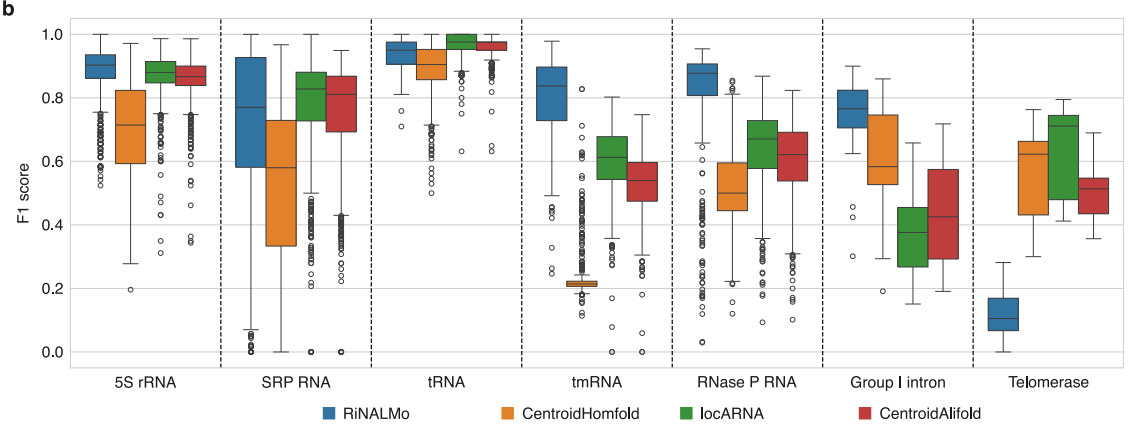


Fig. 5 | Inter-family secondary structure prediction for RiNALMo and homology-based tools. a Secondary structure prediction average F1 scores for the ArchiveII evaluation datasets. The numbers in brackets next to each RNA family name represent the count of RNAs for which at least one homolog was identified, followed by the total number of RNAs in that dataset. RNAs for which no homologs were found are ignored. The best result for each evaluation dataset in the tables is shown in bold. **b** Distribution of secondary structure prediction F1 scores for different tools on the ArchiveII evaluation datasets (sample sizes *n*: 1278, 738, 510, 456, 442, 37, and 35, respectively). Box plots show the median (center line), 25th and 75th percentiles (bounds of box), whiskers extending to the smallest and largest values within 1.5× the interquartile range, and individual outliers beyond the whiskers. Minimum and maximum values are 0 and 1, respectively.

of exons to form mature mRNAs (see Fig. 6a). Precisely pinpointing splice sites-the donor and acceptor sites that mark the boundaries between exons and introns, and vice versa-is essential for accurately predicting gene structure and location.

Identifying the splice sites can be cast as a binary sequence-level classification task. A widely used dataset of positive and negative subsets of splice-site sequences was proposed in ref. 56. The dataset was constructed by randomly selecting sequences from the exon/intron regions of the G3PO+ genomic sequences[57]. The dataset consists of error-free splice-site sequences from a diverse set of 148 eukaryotic organisms, including humans. The test dataset consists of four different species not seen in the training dataset.

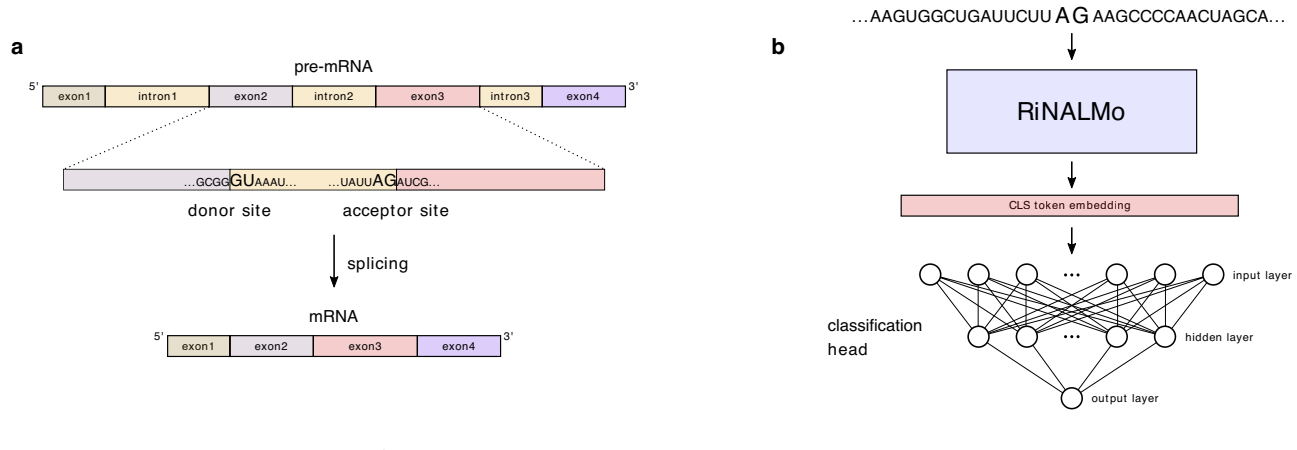
We separately fine-tuned the model, first for donor and then for acceptor splice-site prediction. The splice-site prediction pipeline using RiNALMo embeddings is illustrated in Fig. 6b. Finally, we compared our model's performance with other RNA LMs RNA-FM[21] and Uni-RNA[22], and several established methods such as Spliceator[56] and SpliceBERT[26]. We present the results in Fig. 6c. Separate results for donor and acceptor splice-site prediction can be found in Supplementary Tables S6 and S7, respectively. Figure 6c reports the average value of donor and acceptor prediction results. We fine-tuned other models and used the same prediction head if they were publicly available and easy to fine-tune. Our fine-tuned model outperforms other models, showing its powerful generalization properties. Notice that RiNALMo even outperforms SpliceBERT, an LLM pre-trained exclusively on pre-mRNA sequences. More details on the splice-site prediction task and the model's hyperparameters can be found in the Multi-Species Splice-Site Prediction section and Supplementary Note 4.

Non-coding RNAs are RNA molecules that play vital regulatory roles in a wide range of biological processes. Among the most abundant and functionally significant ncRNAs are transfer RNAs and ribosomal RNAs (rRNAs), both playing an important part in protein synthesis, microRNAs which are essential in regulating gene expression, small nuclear RNAs involved in the processing and splicing of pre-mRNA, etc.

Determining the family of ncRNA is a multiclass classification task. We utilized RiNALMo to predict short noncoding RNA functional families from Rfam[58] using the sequence as an input. The dataset and data preprocessing were adopted from ref. 59. After data preprocessing, the dataset comprised ncRNAs shorter than 200 nucleotides arranged in 88 different Rfam families. We assessed the classification performance of RiNALMo for the original ncRNA sequences, for which we denoted the experiment as 0% boundary noise, and sequences with random nucleotides, equivalent to 100% of the sequence length, added at both ends of the original sequence. Random parts maintained the same single-nucleotide and di-nucleotide frequencies as the original sequence. The second experimental setup we denoted as 200% boundary noise. This way, we introduced the uncertainty of where the ncRNA sequence starts and ends. Adding random nucleotides to the original ncRNA sequences was also adopted from ref. 59.

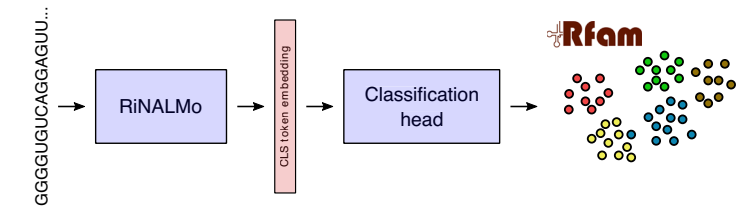
We fine-tuned RiNALMo separately on datasets with 0% and 200% boundary noise. The ncRNA functional family classification pipeline using RiNALMo embeddings is illustrated in Fig. 6d. We compared RiNALMo's performance against other RNA LMs Uni-RNA[22] and RNA-FM[21], as well as against CNN-based methods with different sequence representation approaches proposed by ref. 59. We present the results in Fig. 6e. We fine-tuned RNA-FM and used the same prediction head as for RiNALMo. Our fine-tuned model outperforms other state-of-the-art models and shows robustness against boundary noise. This further validates the ability of RiNALMo to extract evolutionary information. More details on the ncRNA classification task and the model's

https://doi.org/10.1038/s41467-025-60872-5



c

Model	FT	Fish	Fly	Plant	Worm
RiNALMo	✓	**97.70**	**96.11**	**96.25**	**95.63**
SpliceBERT	✓	96.70	95.83	94.96	94.82
Uni-RNA	×	96.35	94.98	93.62	93.94
RNA-FM	✓	96.15	94.46	92.08	92.09
Spliceator	×	95.10	94.80	94.90	94.10



e

Model	FT	0% boundary noise	200% boundary noise	Average
RiNALMo	✓	**0.990**	**0.987**	**0.989**
Uni-RNA	×	0.985	0.984	0.985
RNA-FM	✓	0.915	0.959	0.937
3-mer	×	0.890	0.840	0.865
2-mer	×	0.880	0.810	0.845
1-mer	×	0.870	0.810	0.840

Fig. 6 | RNA functional classification tasks. *Splice-site prediction*. **a** A pre-mRNA transcript consists of non-coding, i.e., introns, and coding regions, i.e., exons. Introns are located between two exons of a gene. As part of the RNA processing pathway, introns are removed by cleavage at splice sites. These sites are found at 5′ and 3′ ends of introns, known as donor and acceptor splice sites, respectively. Most frequently, the 5′ end of introns begins with the dinucleotide GU, and the 3′ end of introns ends with AG. **b** An input to RiNALMo is a 400-nucleotide-long RNA sequence from the GS_1 dataset. We utilize only the CLS embedding that then passes through a two-layer MLP classification head. The output layer gives information on whether a sequence contains a donor/acceptor site or not. **c** Classification F1 score for splice-site prediction. Here, we report the average value of donor and acceptor prediction results. *ncRNA family classification*. **d** Given an RNA sequence the goal is to classify its ncRNA family. The procedure is again similar to the procedure in (**b**): Original and noisy RNA sequences from the Rfam dataset are input to RiNALMo. We utilize only the CLS embedding that then passes through a two-layer MLP classification head. The output layer determines which of the 88 Rfam families the input ncRNA belongs to. **e** ncRNA family classification accuracy for noiseless and noisy input sequences and the average accuracy. In (**c** and **e**), FT denotes whether we fine-tuned the model or represented direct citations from the original papers with the same split train/test datasets. The best result for each evaluation dataset is shown in bold.

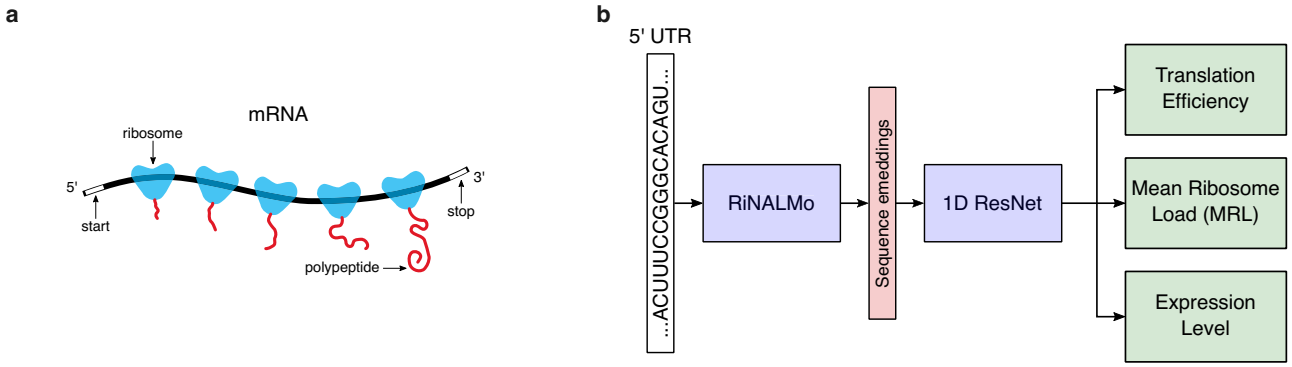
hyperparameters can be found in the Multi-Species Splice-Site Prediction section and Supplementary Note 4.

Fine-tuning RiNALMo for mRNA translation downstream tasks

Due to efficiency reasons, cells usually use groups of multiple ribosomes to translate the mRNA. These groups are called polyribosomes, enabling the cell to create multiple proteins from a single mRNA. To quantify protein synthesis activity, an MRL metric, defined as the average number of ribosomes that translate the mRNA instructions into polypeptides, has been introduced. Several other metrics describe the translation of mRNA to polypeptides and protein production. The mRNA TE quantifies the rate of translation into proteins and the mRNA EL indicates the relative abundance of the mRNA transcript in the cell. These quantities are predictive of the 5′ untranslated region (UTR) of mRNAs. Figure 7a illustrates the translation of an mRNA to polypeptides.

MRL, TE, and EL prediction tasks can be viewed as regression tasks where the input is the 5′ UTR region of the mRNA. Commonly used datasets of 5′ UTR sequences with measured MRL values are provided by ref. 60. Two evaluation datasets, namely Random7600 and

https://doi.org/10.1038/s41467-025-60872-5



c

Model	FT	Random7600	Human7600
RiNALMo	✓	**0.92**	**0.86**
Uni-RNA	×	0.91	0.85
RNA-FM	✓	0.84	0.79
Optimus 5-Prime	✓	0.84	0.78

d

Model	FT	Muscle	PC3	HEK	AVG
RiNALMo	✓	**0.72**	**0.70**	**0.66**	**0.69**
UTR-LM	×	0.67	0.65	0.60	0.64
RNA-FM	✓	0.62	0.60	0.55	0.59
Cao-RF	×	0.63	0.63	0.55	0.60
Optimus 5-Prime	×	0.41	0.38	0.36	0.38

e

Model	FT	Muscle	PC3	HEK	AVG
RiNALMo	✓	**0.70**	**0.67**	**0.69**	**0.69**
UTR-LM	×	0.66	0.63	0.65	0.65
RNA-FM	✓	0.59	0.49	0.51	0.53
Cao-RF	×	0.64	0.59	0.57	0.60
Optimus 5-Prime	×	0.15	0.19	0.18	0.17

Fig. 7 | Mean ribosome loading, translation efficiency, and expression level prediction. a To produce multiple proteins from a single mRNA strand, cells utilize multiple ribosomes. The mean ribosome load metric (MRL), calculated based on a 5' UTR sequence, measures the number of ribosomes translating an mRNA at a given time. Further, 5' UTR is also predictive of the mRNA translation efficiency (TE) that quantifies the rate of translation into proteins and the mRNA expression level (EL) that reflects the relative quantity of the mRNA transcript in the cell. **b** 5' UTR representation produced by RiNALMo is forwarded to the one-dimensional convolutional ResNet which outputs the MRL, TE, or EL prediction. **c,** R^2 score for MRL prediction on the Random7600 and Human7600 datasets. **d** Average Spearman correlation coefficient across ten folds for TE prediction for human muscle tissue (Muscle), human prostate cancer cell (PC3), and human embryonic kidney 293T (HEK), and the average across cell lines and tissue types. **e** Average Spearman correlation coefficient across ten folds for EL prediction for the same datasets as for the TE downstream task. In (**c, d**, **e**) FT denotes whether we trained the model or represented direct citations from the original papers with the same split train/test datasets. The best result for each evaluation dataset is shown in bold.

Human7600, were created by sampling the original dataset containing human and random UTR sequences. For the TE and EL prediction downstream tasks, we followed the work reported by ref. 30. Three endogenous human 5′ UTR datasets analyzed by ref. 61 were used. Each dataset originated from a distinct cell line or tissue type: human muscle tissue (Muscle), human prostate cancer cell (PC3), and human embryonic kidney 293T (HEK). Each sequence of these three datasets provides measurements of translation efficiency and expression level.

We separately fine-tuned RiNALMo to predict the MRL values, TE, and EL for the 5′ UTR sequences from datasets. The prediction pipeline is illustrated in Fig. 7b. RiNALMo outputs are fed into a prediction head consisting of six ResNet[62] blocks. The mean squared error was used as the loss function for MRL prediction. In TE and EL downstream tasks, we used the Huber loss function.

For the MRL downstream task, the model's performance was compared to other RNA LMs Uni-RNA[22] and RNA-FM[21]. We also compared our model to the popular Optimus 5-prime model[60] specialized for MRL prediction. R^2 was used as the evaluation metric, and the results are reported in Fig. 7c. Fine-tuned RiNALMo outperforms other models. Notice that RiNALMo can generalize on human UTRs despite being fine-tuned only on sequences of random origin, again proving its generalization capability. More details on MRL prediction can be found in the Multi-Species Splice-Site Prediction section and Supplementary Note 4.

For the TE and EL prediction, RiNALMo's performance was compared to other LMs UTR-LM[30] and RNA-FM[21]. Uni-RNA[22] was not evaluated on the TE and EL prediction downstream tasks. We also compared RiNALMo to a random forest method proposed in ref. 61 and again to the popular Optimus 5-prime model[60]. The Spearman correlation coefficient was used as the evaluation metric. In Fig. 7d and Fig. 7e, we report the average Spearman correlation coefficient across ten folds for TE and EL tasks, respectively. From the tables, it can be seen that the fine-tuned RiNALMo outperforms other models on all the evaluation datasets. More details on TE and EL prediction can be found in the Translation Efficiency and Expression Level Prediction section and Supplementary Note 4.

Fine-tuning the model for mRNA translation-related downstream tasks and the achieved state-of-the-art results demonstrate good generalization capabilities of RiNALMo to other types of RNAs. Namely, RiNALMo was pre-trained on ncRNAs without seeing a single mRNA or its UTR parts. Despite this, it outperforms other general-purpose RNA language models and UTR-LM which was specifically pre-trained on 5′ UTR sequences and designed for mRNA translation-related tasks.

Discussion

We pre-trained our RNA language model on a vast amount of ncRNA sequences from several databases, including RNAcentral, Rfam, nt, and Ensembl. The data was carefully curated to ensure sequence diversity in each training batch, which subsequently led to better generalization capabilities of the pre-trained language model. We used t-SNE on the classification token embeddings to show that RiNALMo's output representations contain information about the RNA family and that the RNAs with similar structures are close in the model's embedding space. A more insightful way of assessing the embeddings' expressiveness and whether the model captures hidden structure information is to assess it on downstream tasks.

First, we fine-tuned and tested RiNALMo on two secondary structure prediction tasks. The first task was an intra-family secondary structure prediction, where RNAs from the same family came in both the training and test datasets. The results showed that fine-tuned RiNALMo's output representation embeds important structure information about the input sequence and, when utilized with a proper prediction head, leads to state-of-the-art performance. The second task was an inter-family secondary structure prediction, where an RNA family from the test dataset was not seen in the training dataset. It was shown previously that deep learning methods do not generalize well across RNA families[47]. However, RiNALMo outperformed both thermodynamics-based and deep learning methods and showed that RNA language models can generalize well on unseen RNA families. This demonstrates the outstanding generalization capability of RiNALMo for secondary structure prediction tasks.

Furthermore, we fine-tuned RiNALMo on five function-related RNA downstream tasks. Four of these tasks were related to the pre-mRNA or the untranslated region of the mRNA whose examples were not contained in the pre-training dataset. The "Results" section showed that RiNALMo again generalizes well and can capture important functional information from RNA sequences from previously unseen RNA types. RiNALMo outperforms other general-purpose RNA language models and language models trained on RNA types specific for those downstream tasks, such as SpliceBERT[26] and UTR-LM[30].

In future work, we will explore the augmentation of the pre-training dataset by adding coding RNAs and evaluate how it affects the model performance on the structural and functional downstream tasks. Having bigger data might require a larger RNA LM which we also have in consideration. Multimodal pre-training, including sequence and chemical mapping data, is another direction for improving the representations for structure-related downstream tasks, which we will explore in the future. Finally, we plan to employ RiNALMo in several other structure-related tasks. Of particular interest would be testing RiNALMo on tertiary structure prediction tasks and seeing whether its expressive output embeddings and generalization capability can improve the performance of existing or new prediction tools. Another interesting application would be employing RiNALMo's embeddings for sequence conditioning for RNA design[63,64].

To conclude, we presented RiNALMo, a new largest-to-date general-purpose RNA LM pre-trained on a dataset of 36M ncRNA sequences using MLM. We showed that by pre-training on a carefully curated RNA dataset and using the most modern Transformer techniques, an LM can capture hidden knowledge and important structural information from unlabeled RNA sequences. The results on downstream tasks proved the generalization capability of RiNALMo and the expressiveness of its output representations. Of particular importance are the results of the inter-family secondary structure prediction task, where we showed that RiNALMo can generalize well on RNA families unseen during fine-tuning, unlike other DL methods. Due to the significance of the results, we believe RiNALMo presents a valuable asset for advancing our understanding of RNA structures and functions.

Methods

RNA language model

RiNALMo is an encoder-only Transformer. An input RNA sequence is tokenized, turned into a 1280-dimensional vector using a learned input embedding model and passed to the Transformer. RiNALMo consists of 33 Transformer blocks, and each block comprises a multi-head attention and a feed-forward network (FFN). The position of the tokens is encoded using the RoPE[31], which effectively encapsulates both the relative and the absolute positional information. Each multi-head attention has 20 attention heads. Multi-head attention is illustrated in Supplementary Fig. S1. To improve the pre-training efficiency of such a large model, we employed the IO-aware FlashAttention-2[33], a fast and memory-efficient exact attention. In the FFN, we use two linear layers and the SwiGLU activation function[32] which combines the advantages of the Swish activation function and the gated linear unit (GLU). The FFN layers have hidden size $d_{ff} = 3413$, scaling the model to 650M parameters. The Transformer modules are interconnected using residual connections. We use the layer normalization put inside the residual blocks to stabilize the training and have well-behaved gradients at initialization[65]. We did an ablation study for RoPE and the SwiGLU activation function which showed the effectiveness of these techniques on the downstream tasks compared to the conventional absolute positional encoding and GELU activation function used in RNA-FM. The ablation study results are provided in Supplementary Tables S14 and S15.

Tokenization

In this work, each nucleotide is a single token. During tokenization, we replace all "U"s in the sequences with "T"s. This leads to a vocabulary involving standard IUPAC nucleotide codes ("A", "C", "T", "G", "R", "Y", "K", "M", "S", "W", "B", "D", "H", "V", "N", and "-") and a special token for inosine ("I"). We use additional tokens commonly used in MLM, such as `[CLS]`, `[EOS]`, `[PAD]`, and `[MASK]`. During masking, we change nucleotides with the `[MASK]` token or replace it with one of the four main types of nucleotides from the vocabulary or the "any nucleotide" ("N") token. Tokens `[CLS]` and `[EOS]` are added at the beginning and end of the sequence. The `[PAD]` token is appended at the end of shorter sequences to have all the sequences in a batch of the same length.

Data preprocessing

For pre-training dataset preprocessing, we implemented a multi-step preparation pipeline. First, we collected noncoding RNA sequences from publicly available datasets RNAcentral, nt, Rfam and Ensembl. We removed sequences shorter than 16 and longer than 8192. Furthermore, we also removed sequence duplicates with *seqkit rmdup* and the resulting unique sequences were clustered with *mmseqs easy-linclust* with options *--min-seq-id 0.7* and *-c 0.8*. Finally, we saved the processed dataset into *LMDB* (Lightning Memory Mapped Database) so we could easily and quickly access any data sample during the pre-training. In the end, the dataset consisted of 36M unique ncRNA sequences clustered into 17M clusters. In comparison, ref. 21 collected ncRNAs from the RNAcentral database only. As a pre-processing step, the authors removed only duplicate RNA sequences. We clustered the RNA sequences to ensure sequence diversity in each training batch since we later sample from each cluster once per epoch.

Pre-training

We pre-trained RiNALMo using the MLM task where we corrupted unlabeled RNA sequences and then tasked the model to reconstruct them. To corrupt the input sequence, we randomly selected 15% of the tokens in the training sequence. Of these, 80% are masked, i.e., replaced with the unique vocabulary token `[MASK]`, 10% are replaced with a randomly selected token from the vocabulary, and the remaining 10% are left intact.

Let us denote the corrupted sequence by $\tilde{\boldsymbol{X}} = \{\tilde{x}_i\}$. We train RiNALMo parameterized by θ to reconstruct $\boldsymbol{X}$ by predicting the masked tokens conditioned on $\tilde{\boldsymbol{X}}$. For a given input token $\tilde{x}_i$, the loss is the probability of the correct nucleotide, given $\tilde{\boldsymbol{X}}$:

$$\mathcal{L}_{MLM} = -\log p_\theta(\tilde{x}_i = x_i|\tilde{\boldsymbol{X}}). \quad (1)$$

The weight updates are based on the average loss over the sampled tokens from a single training sequence (or a batch of sequences):

$$\mathcal{L}_{MLM} = -\frac{1}{|\mathcal{M}|}\sum_{i\in\mathcal{M}} \log p_\theta(\tilde{x}_i = x_i|\tilde{\boldsymbol{X}}), \quad (2)$$

where $\mathcal{M}$ is the index set of masked tokens.

Due to the computational limitations, during training, we limited the length of the input into the language model to 1024 tokens. Every sequence begins with a `[CLS]` token, i.e., a classification token, and ends with an `[EOS]` token, i.e., an end-of-sequence token. If an input sequence is longer than 1022 nucleotides, a random part of the sequence will be cropped out and fed into the model during each epoch, similar to what was done in ref. 13. To implicitly give information to the model that the sequence was cropped, it does not begin with the `[CLS]` token nor ends with the `[EOS]` token unless we crop the beginning or end of the sequence.

To ensure sequence diversity in each training batch during pre-training, we randomly sampled each sequence in the batch from a different sequence cluster. Effectively, it means that in every epoch RiNALMo saw 17M samples, i.e., one sequence from each of the clusters, and in each epoch, sequences were sampled randomly from the clusters using a new seed.

We pre-trained RiNALMo with seven A100 GPUs of 80 GB memory for 2 weeks. The batch size was set to 192 per GPU. We adopted the cosine annealing learning rate schedule with a linear warm-up. During the warm-up period, learning rate increases from 10^{-7} to 5×10^{-5} for 2000 steps. For the cosine annealing schedule, the minimum learning rate was set to 10^{-5}. The gradient norm was clipped to 1.0.

We pre-trained several configurations of the LM: RiNALMo-650M with 650M parameters, RiNALMo-150M with 148M parameters, and RiNALMo-33M with 33.5M parameters. We did several experiments to evaluate how the model's size influences its performance on the pre-training and downstream tasks. First, we measured perplexity. Perplexity is a metric used to evaluate how well an LM predicts a sample of text or, in our case, an RNA sequence. The perplexity is defined as the exponential of the negative log-likelihood of the sequence. To efficiently calculate the perplexity, we used the following term

$$\text{PERPLEXITY}(x) = \exp\{-\log p_\theta(\tilde{x}_{i\in\mathcal{M}} = x_i|\tilde{\boldsymbol{X}})\}. \quad (3)$$

Lower perplexity means the LM is better at reconstructing the masked tokens in the MLM pre-training task.

In order to evaluate how the number of parameters affects the model's ability to reconstruct the masked tokens, we calculated the perplexity of RiNALMo for several different model sizes. Validation perplexity was measured on a 1% random-split holdout of the pre-training dataset. The perplexity values are given in Supplementary Fig. S2. It can be seen that as we increase the model size, the perplexity decreases, meaning the larger the LM gets, the more capable it is of reconstructing the missing tokens.

The hyperparameters of RiNALMo models of different sizes are given in Supplementary Table S1, and the comparison of their performance on the downstream tasks are given in Supplementary Tables S8, S9, S10, S11, S12, and S13. Based on these results, we can conclude that besides novel techniques such as RoPE and SwiGLU, model size is important in achieving a boost in performance on pre-training and consequently downstream tasks.

Secondary structure prediction

To adapt RiNALMo for the secondary structure prediction, we fine-tuned the model with a simple binary classification task with binary cross-entropy loss, where we tasked the model to classify each nucleotide pair as either paired or unpaired. A simple prediction head was attached to the language model and was fed with nucleotide pair representations. The prediction head was a bottleneck residual neural network (ResNet) with two convolutional blocks. These representations are obtained by applying outer concatenation to RiNALMo's output. For example, to obtain the vector representation of the nucleotide pair (i, j), we simply concatenate the representation of the nucleotide j to the representation of the nucleotide i.

In the end, our secondary structure prediction pipeline outputs a matrix where each element represents a pairing probability logit for a certain nucleotide pair. Because of the symmetry of secondary structures (if nucleotide i is paired with nucleotide j, then j is paired with i as well), we calculate training loss only on matrix elements "above" the main diagonal.

During fine-tuning, we utilized gradual parameter unfreezing. After every three epochs, we unfroze an additional three RiNALMo's layers. In the first three epochs, we trained only the prediction head. The model was fine-tuned for 15 epochs and the learning rate was initially set to 10^{-4} with a linear decay schedule. We used the same prediction head architecture for RNA-FM fine-tuning. Due to architectural differences between RiNALMo and RNA-FM, we decided to modify the fine-tuning schedule. The prediction head was first pre-trained while RNA-FM parameters were kept frozen. After three epochs we unfroze RNA-FM and fine-tuned it alongside the prediction head.

To convert base pairing probabilities to a proper secondary structure, we implemented a simple greedy approach where we iteratively set nucleotide pairs with the highest pairing probability as paired and then excluded all possible clashing pairs (pairs where the same nucleotide is paired with multiple other nucleotides) from being set as paired in future iterations of the algorithm. During this procedure, we ignore non-canonical nucleotide pairings and pairings that would cause a "sharp" hairpin loop (i-th nucleotide cannot be paired with the j-th nucleotide if $|i-j|<4$). The classification threshold was tuned on the validation set to ensure a balanced pairing ratio.

When assessing the performance of secondary structure prediction, it is important to consider RNA structural dynamics, and that's why it is helpful to view predictions that are close enough as correct. Same as the other methods we compared RiNALMo to, we employed the metric calculation approach proposed in ref. 50. To be more precise, for a nucleotide pairing (i, j) where i and j represent nucleotide indices in the RNA sequence, $(i\pm1, j)$ and $(i, j\pm1)$ pairings are also considered correct predictions. To obtain the F1 scores we reported in this study, we calculated the F1 score separately on each structure and then averaged those values. Again, for the sake of fairness, notice that we have reported F1 scores for all other methods and tools using the exact same calculation.

In the RNA community, it is also customary to use the interaction network fidelity (INF) measure[66], which is tailored for evaluating RNA base pairs. The INF measure provides a more balanced assessment of the predictive algorithm's performance than the F1 score across both positive and negative classes. Therefore, and for the completeness of our study, we computed INF scores for the secondary structure prediction evaluation datasets, which are provided in Supplementary Tables S3, S4, and S5. Regarding the INF measure, RiNALMo still outperforms other methods we used for the comparison.

In Supplementary Table S2, we report sequence-length weighted F1 scores on the inter-family secondary structure prediction task. We also provide the performance of smaller LMs, the ablation study and the impact of fine-tuning in Supplementary Note 4.

Secondary structure visualizations shown in the figures were generated using *forna*[67] and RNAstructures's *draw*[34].

Multi-species splice-site prediction

Splice-site prediction task is essentially a binary sequence-level classification task, where a method has to detect whether a query RNA sequence contains a donor/acceptor site or not. We fine-tuned RiNALMo on the training dataset to predict splice sites in the sequences. The classification token (CLS) was used as input to the classification head. This head is configured as a two-layer multilayer perceptron (MLP) featuring a hidden layer with 128 neurons and employing the GELU activation function. Cross-entropy loss served as our choice for the loss function. We separately fine-tuned the model, first for the donor and then for the acceptor splice-site prediction task.

We used the dataset denoted as GS_1 in the paper, which was proposed in ref. 56. GS_1 has the same ratio of positive to negative samples, whose negative samples consist of exon, intron, or false-positive sequences. The length of the input sequences was set to 400 nucleotides for both the training and test datasets. The training dataset for the acceptor splice-site prediction consists of 22,178 samples and the training dataset for the donor splice-site prediction comprises 21,973 samples. Each test dataset for both donor and acceptor splice-site prediction consists of 20,000 samples.

The prediction head was uniformly initialized from $\mathcal{U}(-\sqrt{1/d}, \sqrt{1/d})$, where d is the input feature dimension. The learning rate was set to 10^{-5} and the language model with the prediction head was fine-tuned for two epochs when it achieved the best results on the validation dataset. The batch size was set to 32.

We fine-tuned the SpliceBERT model and the RNA-FM model in combination with a two-layer MLP prediction head. We fine-tuned the models on the same training/validation dataset used for fine-tuning RiNALMo with the same learning rate.

The performance of smaller LMs and the impact of fine-tuning for the splice-site prediction task can be found in Supplementary Table S10.

ncRNA family classification

The ncRNA functional family classification downstream task is a sequence-level multiclass classification task. We separately fine-tuned RiNALMo on the two training datasets. The CLS was used as input to the classification head. The head was configured as a two-layer MLP featuring a hidden layer with 256 neurons and employing the GELU activation function. Cross-entropy loss served as our choice for the loss function.

The noiseless and noisy datasets were constructed following the preprocessing reported by ref. 59. Starting from the original Rfam dataset[58], the procedure excludes classes annotated as long ncRNAs and with an average sequence length greater than 200 nucleotides. This leads to a dataset with 371,619 sequences among 177 Rfam families. Furthermore, the procedure removes families whose clustering was highly correlated with sequence length and families with less than 400 RNA sequences to ensure data quality. This dataset consisted of 306,016 sequences distributed among 88 different Rfam classes. Each Rfam family was randomly split into three subsets: training (84%), validation (8%), and test (8%). The procedure ensured that all sequences in the validation and test sets have a similarity, in terms of normalized Hamming distance, less than 0.5 with any other sequence in the training set. This is to limit the potential bias arising from an over-representation of highly similar homologous sequences in random splits. Finally, the training and validation sets were sampled with replacements to address the imbalanced class distribution. The final dataset contained 105,864 training, 17,324 validation and 25,342 test sequences.

The classification head was uniformly initialized from $\mathcal{U}(-\sqrt{1/d}, \sqrt{1/d})$, where d is the input features dimension. We trained the model using an AdamW optimizer with a weight decay of 0.01. The learning rate was set to 8×10^{-6} and the language model with the classification head was fine-tuned for 25 epochs. The batch size was set to 128. The final model weights were chosen based on the best accuracy achieved on the validation set.

We fine-tuned RNA-FM in combination with a two-layer MLP prediction head. The best classification performance it achieved was with 128 neurons in the hidden layer and 10^{-5} learning rate. We fine-tuned the models on the same training/validation dataset used for fine-tuning RiNALMo.

In Supplementary Table S11, we provide the results of smaller LMs and the impact of fine-tuning on the ncRNA family classification task.

Mean ribosome loading prediction

We fine-tuned RiNALMo on the appropriate training dataset to predict the MRL value for the given 5′ UTR sequence. Language model outputs are fed into a prediction head which consists of six ResNet convolutional blocks. The mean squared error was used as the loss function. RNA-FM was fine-tuned with the same prediction head as RiNALMo.

Each block of the prediction head consists of two 1D convolution layers followed by instance normalization and ELU activation function. Parameters of the prediction head were initialized with the default Pytorch[68] parameter initialization method.

Training and evaluation datasets were produced with the procedure described in ref. 60. Two evaluation datasets were created by sampling the original dataset which contains 83,919 human and random UTR sequences of varying lengths. To ensure that each sequence length is represented equally in the dataset, 100 sequences with the deepest read coverage were selected for every length from 25 to 100 nucleotides. The same approach has been applied to both the human and random 5′ UTR sequences, yielding two evaluation datasets of size 7600 called Random7600 and Human7600. All remaining random 5′ UTRs with acceptable read coverage depth were used as the training dataset.

All MRL targets were standardized with the mean and standard deviation of training MRL values.

The model was fine-tuned for 50 epochs. In the first 5 epochs of the training, only the prediction head was trained. The learning rate was set to 10^{-4} and linearly decayed to 10^{-5} over the first 5000 training steps after which it remained constant. The batch size was set to 64. RNA-FM was fine-tuned with the same training procedure.

The performance of smaller LMs and the impact of fine-tuning for the MRL prediction task can be found in Supplementary Table S12.

Translation efficiency and expression level prediction

Fine-tuning and prediction of mRNA TE and EL from 5′ UTR sequences followed the procedure for the MRL prediction downstream task. Language model outputs are fed into a prediction head which consists of six ResNet convolutional blocks. The Huber loss was used as the loss function. The architecture and the initialization of the prediction head were the same as for the MRL prediction task. RNA-FM was fine-tuned with the same prediction head as RiNALMo.

Evaluation data were the same as reported in the UTR-LM paper[30]. The data consisted of three datasets, namely human muscle tissue, human prostate cancer cell line PC3, and human embryonic kidney 293T cell line. The Muscle, PC3, and HEK datasets contained 1257, 12,579, and 14,410 5′ UTR sequences, respectively. Following ref. 30 and to achieve a fair comparison, only a fixed 5′ UTR length of 100 nucleotides was chosen for training. These 100 nucleotides are located upstream of the coding region in the mRNA. Following ref. 30, for training and testing we used tenfold cross-validation for each cell line for both TE and EL prediction tasks.

In the datasets, mRNA expression level was determined using RNA-seq reads per kilobase of transcript per million mapped reads (RPKM), and translation efficiency for each transcript was calculated by dividing the Ribo-seq RPKM by the RNA-seq RPKM. The translation efficiency and expression level labels are in the natural logarithm space.

The model was fine-tuned for 30 epochs. In the first 5 epochs of the training, only the prediction head was trained. The learning rate was set to 8×10^{-5} and linearly decayed to 8×10^{-6} over the first 3000 training steps after which it remained constant. The batch size for the Muscle dataset was set to 8, and for the PC3 and HEK datasets to 32. RNA-FM was fine-tuned with the same training procedure.

In Supplementary Table S13, we provide the results of smaller LMs and the impact of fine-tuning on the TE and EL prediction task.

Reporting summary

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

Data availability

We used several databases with unannotated RNA sequences, namely RNAcentral[24], nt[69], Rfam[45] and Ensembl[70]. The RNAcentral dataset is available at https://ftp.ebi.ac.uk/pub/databases/RNAcentral/releases/22.0/sequences/. The nt dataset is available at https://ftp.ncbi.nlm.nih.gov/blast/db/FASTA/. The Rfam 14.9 dataset is available at https://ftp.ebi.ac.uk/pub/databases/Rfam/14.9/fasta-files/Rfam.fa.gz. Finally, the Ensembl dataset is available at https://ftp.ensembl.org/pub/release-109/fasta/and https://ftp.ensemblgenomes.ebi.ac.uk/pub/bacteria/release-56/.The intra-family RNA secondary structure dataset with train/test splits is available at https://dl.dropboxusercontent.com/s/w3kc4iro8ztbf3m/bpRNA_dataset.zip. Structurally dissimilar datasets TrainSetA and TestSetB are available at https://github.com/mxfold/mxfold2/releases/download/v0.1.1/Rivas.tar.gz. The inter-family secondary structure dataset consists of nine families and train/test splits are available at https://github.com/marcellszi/dl-rna/releases/download/Data/ct-splits.tar.gz. For the multi-species splice-site downstream task, we used the dataset denoted as GS_1 in ref. 56. The dataset is available at https://git.unistra.fr/nscalzitti/spliceator and https://zenodo.org/records/7995778. The dataset[59] used in the ncRNA family classification task can be found at https://github.com/bioinformatics-sannio/ncrna-deep/tree/master/datasets/Rfam-novel. The mean ribosome loading dataset[60] used in the paper can be found at https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE114002&format=file. The mRNA translation efficiency and expression level datasets reported in ref. 61 and used in ref. 30 as well as in this paper are available at https://drive.google.com/drive/folders/190oihtrwCxWjtDCK9kJzyhXPKxbr5xoR and https://codeocean.com/capsule/4214075/tree/v1.

Code availability

The code repository[71] is available on https://github.com/lbcb-sci/RiNALMo and the pre-trained and fine-tuned weights are available on https://zenodo.org/records/15043668. The weights can be automatically downloaded using the script provided in the repository. We provide scripts for fine-tuning the pre-trained model on the downstream tasks from the "Results" section. The data used in the downstream tasks can be automatically downloaded and preprocessed using the scripts in the code repository.

References

1. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long and Short Papers), (Minneapolis, Minnesota, USA, 2019).
2. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. Improving language understanding with unsupervised learning. OpenAI https://openai.com/index/language-unsupervised/ Accessed 9 Dec 2024 (2018).
3. Raffel, C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21**, 5485–5551 (2020).
4. Brown, T. et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **33**, 1877–1901 (2020).
5. Touvron, H. et al. Llama: open and efficient foundation language models. Eprint http://arxiv.org/abs/2302.13971 arXiv:2302.13971 (2023).
6. Chowdhery, A. et al. PaLM scaling language modeling with pathways. *J. Mach. Learn. Res.* **24**, 1–113 (2023).
7. Ferruz, N., Schmidt, S. & Höcker, B. ProtGPT2 is a deep unsupervised language model for protein design. *Nat. Commun.* **13**, 4348 (2022).
8. Madani, A. et al. Large language models generate functional protein sequences across diverse families. *Nat. Biotechnol.* **41**, 1099–1106 (2023).
9. Nijkamp, E., Ruffolo, J. A., Weinstein, E. N., Naik, N. & Madani, A. ProGen2 exploring the boundaries of protein language models. *Cell Syst.* **14**, 968–978 (2023).
10. Wu, R. et al. High-resolution de novo structure prediction from primary sequence. bioRxiv https://doi.org/10.1101/2022.07.21.500999 (2022).
11. Lin, Z. et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).
12. Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
13. Rives, A. et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl Acad. Sci.* **118**, 2016239118 (2021).
14. Heinzinger, M. et al. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinforma.* **20**, 1–17 (2019).
15. Nambiar, A. et al. Transforming the language of life: transformer neural networks for protein prediction tasks. In: *Proc. 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, 1–8 (ACM, 2020).
16. Brandes, N., Ofer, D., Peleg, Y., Rappoport, N. & Linial, M. ProteinBERT: a universal deep-learning model of protein sequence and function. *Bioinformatics* **38**, 2102–2110 (2022).
17. Elnaggar, A. et al. ProtTrans toward understanding the language of life through self-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 7112–7127 (2022).
18. Childs-Disney, J. L. et al. Targeting RNA structures with small molecules. *Nat. Rev. Drug Discov.* **21**, 736–762 (2022).
19. Garner, A. L. Contemporary progress and opportunities in RNA-targeted drug discovery. *ACS Med. Chem. Lett.* **14**, 251–259 (2023).
20. Schneider, B. et al. When will RNA get its AlphaFold moment? *Nucleic Acids Res.* **51**, 9522–9532 (2023).
21. Chen, J. et al. Interpretable RNA foundation model from unannotated data for highly accurate RNA structure and function predictions. https://doi.org/10.48550/arXiv.2204.00300 (2022).
22. Wang, X. et al. Uni-RNA Universal pre-trained models revolutionize RNA research. bioRxiv https://doi.org/10.1101/2023.07.11.548588 (2023).
23. Vaswani, A. et al. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017).
24. RNAcentral Consortium RNAcentral 2021: secondary structure integration, improved sequence search and new member databases. *Nucleic Acids Res.* **49** 212–220 (2020).
25. Zhang, Y. et al. Multiple sequence alignment-based RNA language model and its application to structural inference. *Nucleic Acids Res.* **52**, 3–3 (2024).
26. Chen, K. et al. Self-supervised learning on millions of primary RNA sequences from 72 vertebrates improves sequence-based RNA splicing prediction, *Brief. Bioinform.* **25**, bbae163 (2024).

27. 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).
28. Yang, F. et al. scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data. *Nat. Mach. Intell.* **4**, 852–866 (2022).
29. Celaj, A. et al. An RNA foundation model enables discovery of disease mechanisms and candidate therapeutics. bioRxiv https://doi.org/10.1101/2023.09.20.558508 (2023).
30. Chu, Y. et al. A 5′ UTR language model for decoding untranslated regions of mRNA and function predictions. *Nat. Mach. Intell.* **6**, 449–460 (2024).
31. Su, J. et al. RoFormer enhanced transformer with rotary position embedding. *Neurocomputing* **568**, 127063 (2024).
32. Shazeer, N. GLU variants improve transformer. https://doi.org/10.48550/arXiv.2002.05202 (2020).
33. Dao, T. FlashAttention-2 Faster attention with better parallelism and work partitioning. https://doi.org/10.48550/arXiv.2307.08691 (2023).
34. Reuter, J. S. & Mathews, D. H. RNAstructure software for RNA secondary structure prediction and analysis. *BMC Bioinform.* **11**, 129 (2010).
35. Do, C. B., Woods, D. A. & Batzoglou, S. CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics* **22**, 90–98 (2006).
36. Singh, J., Hanson, J., Paliwal, K. & Zhou, Y. RNA secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. *Nat. Commun.* **10**, 5407 (2019).
37. Fu, L. et al. UFold: fast and accurate RNA secondary structure prediction with deep learning. *Nucleic Acids Res.* **50**, 14–14 (2021).
38. Sato, K., Akiyama, M. & Sakakibara, Y. RNA secondary structure prediction using deep learning with thermodynamic integration. *Nat. Commun.* **12**, 941 (2021).
39. Danaee, P. et al. bpRNA large-scale automated annotation and analysis of RNA secondary structure. *Nucleic Acids Res.* **46**, 5381–5394 (2018).
40. Berman, H. M. et al. The Protein Data Bank. *Nucleic Acids Res.* **28**, 235–242 (2000).
41. Yang, H. et al. Tools for the automatic identification and classification of RNA base pairs. *Nucleic Acids Res.* **31**, 3450–3460 (2003).
42. Rivas, ElenaandLang,RaymondandEddy,SeanR. A range of complex probabilistic models for RNA secondary structure prediction that includes the nearest-neighbor model and more. *RNA* **18**, 193–212 (2011).
43. Andronescu, M., Condon, A., Hoos, H. H., Mathews, D. H. & Murphy, K. P. Efficient parameter estimation for RNA secondary structure prediction. *Bioinformatics* **23**, 19–28 (2007).
44. Andronescu, M., Condon, A., Hoos, H. H., Mathews, D. H. & Murphy, K. P. Computational approaches for RNA energy parameter estimation. *RNA* **16**, 2304–2318 (2010).
45. Kalvari, I. et al. Rfam 14: expanded coverage of metagenomic, viral and microRNA families. *Nucleic Acids Res.* **49**, 192–200 (2020).
46. Wayment-Steele, H. K. et al. RNA secondary structure packages evaluated and improved by high-throughput experiments. *Nat. Methods* **19**, 1234–1242 (2022).
47. 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).
48. Justyna, M., Antczak, M. & Szachniuk, M. Machine learning for RNA 2D structure prediction benchmarked on experimental data. *Brief. Bioinforma.* **24**, 153 (2023).
49. Sloma, MichaelFandMathews,DavidH. Exact calculation of loop formation probability identifies folding motifs in RNA secondary structures. *RNA* **22**, 1808–1818 (2016).
50. Mathews, D. H. How to benchmark RNA secondary structure prediction accuracy. *Methods* **162**, 60–67 (2019).
51. Hamada, M., Sato, K., Kiryu, H., Mituyama, T. & Asai, K. Predictions of RNA secondary structure by combining homologous sequence information. *Bioinformatics* **25**, 330–338 (2009).
52. Will, S., Reiche, K., Hofacker, I. L., Stadler, P. F. & Backofen, R. Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comput. Biol.* **3**, 65 (2007).
53. Hamada, M., Sato, K. & Asai, K. Improving the accuracy of predicting secondary structure for aligned RNA sequences. *Nucleic Acids Res.* **39**, 393–402 (2010).
54. Puton, T., Kozlowski, L. P., Rother, K. M. & Bujnicki, J. M. CompaRNA: a server for continuous benchmarking of automated methods for RNA secondary structure prediction. *Nucleic Acids Res.* **41**, 4307–4323 (2013).
55. Nawrocki, E. P. & Eddy, S. R. Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics* **29**, 2933–2935 (2013).
56. Scalzitti, N. et al. Spliceator: multi-species splice site prediction using convolutional neural networks. *BMC Bioinforma.* **22**, 1–26 (2021).
57. Scalzitti, N., Jeannin-Girardon, A., Collet, P., Poch, O. & Thompson, J. D. A benchmark study of ab initio gene prediction methods in diverse eukaryotic organisms. *BMC Genom.* **21**, 1–20 (2020).
58. Kalvari, I. et al. Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families. *Nucleic Acids Res.* **46**, 335–342 (2018).
59. Noviello, T. M. R., Ceccarelli, F., Ceccarelli, M. & Cerulo, L. Deep learning predicts short non-coding RNA functions from only raw sequence data. *PLoS Comput. Biol.* **16**, 1008415 (2020).
60. Sample, P. J. et al. Human 5′ UTR design and variant effect prediction from a massively parallel translation assay. *Nat. Biotechnol.* **37**, 803–809 (2019).
61. Cao, J. et al. High-throughput 5′ UTR engineering for enhanced protein production in non-viral gene therapies. *Nat. Commun.* **12**, 4138 (2021).
62. 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).
63. Anand, R. ett al. RNA-FrameFlow: flow matching for de novo 3D RNA backbone design. https://doi.org/10.48550/arXiv.2406.13839 (2024).
64. Churkin, A., Barash, D. (eds.) RNA design: methods and protocols. https://doi.org/10.1007/978-1-0716-4079-1 (Springer, 2024).
65. Xiong, R. et al. On layer normalization in the transformer architecture. In: *Proc. International Conference on Machine Learning*, 10524–10533 (PMLR, 2020).
66. Parisien, M., Cruz, J. A., Westhof, E. & Major, F. New metrics for comparing and assessing discrepancies between RNA 3D structures and models. *RNA* **15**, 1875–1885 (2009).
67. Kerpedjiev, P., Hammer, S. & Hofacker, I. L. Forna (force-directed rna): Simple and effective online rna secondary structure diagrams. *Bioinformatics* **31**, 3377–3379 (2015).
68. Paszke, A. et al. Pytorch: an imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **32**, 8024–8035 (2019).
69. Sayers, E. W. et al. Database resources of the National Center for Biotechnology Information in 2023. *Nucleic Acids Res.* **51**, 29–38 (2022).
70. Martin, F. J. et al. Ensembl 2023. *Nucleic Acids Res.* **51**, 933–941 (2022).
71. 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. Zenodo https://doi.org/10.5281/zenodo.15437847 (2025).

Acknowledgements
The authors would like to thank Ivona Martinović for the valuable comments and fruitful discussion on this work. This work was supported in part by the National Research Foundation (NRF) Competitive Research Programme (CRP) under Project Identifying Functional RNA Tertiary Structures in Dengue Virus (NRF-CRP27-2021RS-0001 to Y.W.) and in part by the A*STAR under Grant GAP2: A*STAR RNA-Foundation Model (A*STAR RNA-FM)(I23D1AG079 to M.Š.). The computational work for this article was partially performed on the resources of the National Supercomputing Centre, Singapore (https://www.nscc.sg).

Author contributions
R.J.P. and M.Š. conceived the project. R.J.P. and T.V. designed the method. R.J.P. and T.V. designed and conducted the numerical experiments. R.G.H., Y.W. and M.Š. supervised the study. R.J.P. and T.V. wrote the manuscript. R.G.H., Y.W. and M.Š. provided mentorship and support during the project. All authors approved the manuscript.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-025-60872-5.

Correspondence and requests for materials should be addressed to Mile. Šikić.

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