Journal Title Here, 2023, pp. 1–13

doi: DOI HERE
Advance Access Publication Date: Day Month Year
Paper

PAPER

RNADiffFold: Generative RNA Secondary Structure Prediction using Discrete Diffusion Models

Yizhen Feng,[1,2‡] Zhen Wang,[3,2‡] Qingwen Tian,[4,2] Ziqi Liu,[5,2] Pengju Yan[2,6*] and Xiaolin Li[2,6*]

[1]College of Information Engineering, Zhejiang University of Technology, Hangzhou, 310014, Zhejiang, China, [2]Hangzhou Institute of Medicine, Chinese Academy of Sciences, Hangzhou, 310018, Zhejiang, China, [3]College of Electrical and Information Engineering, Hunan University, Changsha, 410082, Hunan, China, [4]College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, 310014, Zhejiang, China, [5]School of Molecular Medicine, Hangzhou Institute for Advanced Study, University of Chinese Academy of Sciences, Hangzhou, 310024, Zhejiang, China and [6]ElasticMind Inc, Hangzhou, 310018, Zhejiang, China
[‡]These authors contributed equally to this work and should be considered co-first authors.[*]Corresponding author. yanpengju@elasticmind.ai, xiaolinli@ieee.org

Abstract

As a crucial class of macromolecules, RNA plays a vital role in various biological functions within living organisms. Accurately predicting the secondary structure of RNA contributes to a better understanding of its intricate three-dimensional structure and functionality. Previous energy-based and learning-based methods model RNA secondary structures in a static view and impose strong prior constraints. Inspired by the success of diffusion models, in this work, we propose a generative prediction method based on multinomial diffusion. We consider the prediction of contact maps as a pixel-level segmentation task and train the denoise model to iteratively refine contact maps from noise. Additionally, we design an effective condition to extract features from sequences, guiding the model to generate the corresponding secondary structure. These features include sequence one-hot encoding, probability maps from a pre-trained score network, as well as embeddings and attention maps from RNA-FM. Experimental results on both within- and cross-family datasets demonstrate RNADiffFold's competitive performance compared with current state-of-the-art methods. Moreover, RNADiffFold moderately captures dynamic structural features of RNA, as validated on a multi-conformational dataset.

Key words: RNA structure prediction, Deep learning, discrete diffusion model

Introduction

Ribonucleic acid (RNA) is a vital biomolecule exerting significant roles within organisms. Apart from serving as an intermediary for genetic information transfer between DNA and proteins, RNA is involved in multiple functions including catalysis, regulation, and protein synthesis [1]. These functions are predominantly executed by the non-coding regions of RNA. For instance, microRNAs (miRNAs) regulate gene expression at the post-transcriptional level, and their dysregulation is closely associated with the onset and progression of various diseases [2]. Long non-coding RNAs (lncRNAs) extensively participate in mammalian cellular processes such as chromatin modification and transcriptional regulation [3]. As another family of non-coding RNAs, small nuclear RNAs (snRNAs) form spliceosome complexes with proteins, thereby regulating splicing patterns and controlling mRNA abundance [4]. Thus, delving into the functionality of RNA is of paramount importance for elucidating its biological mechanisms.

In cells, RNA typically exists as a single-stranded molecule and folds into specific structures through base pairing via hydrogen bonds, thereby interacting with other biomolecules and exerting functions. The molecular structure of RNA exhibits hierarchy and can be divided into primary, secondary, and tertiary structures. The primary structure consists of a sequence of ribonucleotides composed of four bases. The secondary structure is a two-dimensional topological structure formed by base pairing through hydrogen bonds [5]. Based on this, the tertiary structure further folds into a three-dimensional spatial conformation. Although RNA primarily exerts its biological functions through its tertiary structure, this binding process often relies on motifs such as stems and loops in the secondary structure. However, due to the single-stranded nature of RNA, its tertiary structure is susceptible to environmental influences and exhibits poor stability, making it challenging to obtain directly. Despite significant progress in resolving high-resolution RNA structures through existing experimental methods such as X-ray crystallography, nuclear magnetic resonance (NMR) [6], and cryo-electron microscopy [7], challenges persist in obtaining a sufficient quantity and quality of RNA tertiary structures due to factors such as high experimental costs and resolution limitations. Therefore, deciphering the secondary structure of RNA is crucial for understanding its underlying

© The Author 2023. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

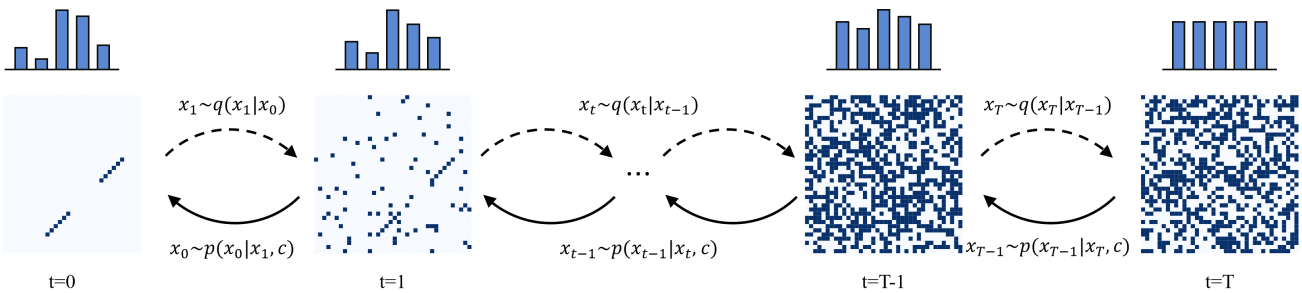

Fig. 1. Overview of RNA secondary structure prediction with multinomial diffusion. During the diffusion process $q(x_t|x_{t-1})$, discrete noise is gradually introduced to corrupt the contact map from left to right. In the denoise process $p_\theta(x_{t-1}|x_t, c)$, a generative model learns to denoise the corrupted map from right to left.

biological mechanisms, representing an essential step from primary sequence to tertiary structure resolution [8].

Over the past decades, numerous researchers have devoted efforts to the study of RNA secondary structure prediction, leading to the development of a comprehensive research framework combining experimental and computational approaches. These methods can extensively be categorized into three major classes: energy-based methods, covariation-based methods, and deep learning-based methods [9]. Energy-based methods primarily utilize experimentally determined parameters to calculate the free energy of RNA structures and identify the most stable secondary structures through dynamic programming [10, 11, 12, 13]. While widely adopted and capable of providing relatively accurate predictions, these methods are limited in that they only consider nested base pairings and struggle with more complex structures such as pseudoknots. Moreover, as the length of RNA increases, the computational complexity of these methods significantly escalates. Covariation-based methods focus on considering the co-evolutionary relationship between RNA sequences and structures to infer secondary structures [14, 15, 16]. These methods can offer highly accurate predictions under certain conditions but may face challenges when dealing with limited information from homologous sequence data. With the continuous accumulation of RNA secondary structure data and the rapid development of deep learning technologies, powerful data support, and technical assurance have been provided for deep learning-based RNA secondary structure prediction methods. Currently, some methods employ neural networks (such as Bi-LSTM, Transformer, Unet, etc.) to calculate base pairing probabilities to capture long-range interactions [17, 18, 19]. Additionally, some research integrates thermodynamic knowledge [20], adopts transfer learning strategies [21], or incorporates evolutionary and mutational coupling information [22] to optimize prediction results and alleviate prediction biases. However, existing deep learning-based RNA secondary structure prediction methods still exhibit shortcomings in terms of model generalization performance, especially when modeling unknown RNA families. This is mainly because their model parameters often stem from a limited pool of known structures, thus limiting the adaptability of the models to new data. In summary, to overcome limitations in training data quantity and distribution and enhance model generalization capabilities, it is necessary to propose new computational methods aiming to achieve more accurate and comprehensive outcomes in the field of RNA secondary structure prediction.

In recent years, diffusion models, as an emerging class of deep generative models, have demonstrated outstanding performance in various prediction tasks [23, 24, 25]. Inspired by this, this study proposes a novel framework for RNA secondary structure prediction based on discrete diffusion models, named **RNADiffFold**. The framework aims to predict a single deterministic RNA secondary structure in a generative manner. RNADiffFold first represents the RNA secondary structure as a binary contact map. The contact map has a size of $L \times L$ (where L is the length of the RNA sequence), and each point in the map can be classified into two categories: 1 denotes pairing, and 0 denotes non-pairing. Based on this representation, a model can be trained to predict the probability of pairing between bases in a given sequence. This approach resembles tasks such as text-to-image translation, where the input sequence can be considered as a "sentence" composed of bases and the contact map can be viewed as an "image" containing information about base pairing. Consequently, the complex task of RNA secondary structure prediction is simplified into a pixel-level image segmentation task.

RNADiffFold primarily consists of two components: the diffusion model component and the conditional control component. The diffusion model component of RNADiffFold is implemented based on discrete data space multinomial diffusion [26]. As illustrated in Figure 1, during the forward diffusion process, the true contact map x_0 is gradually degraded by injecting noise following a uniform categorical distribution. When reaching time step T, x_T transitions into a completely random noise state. In the inverse diffusion process, this study employs UNet [27] as the learning network and adds conditional control to gradually denoise and restore the original contact map. The conditional control component encompasses the sequence information of RNA, including features such as one-hot encoding of the sequence, probability maps from the Ufold scoring network [19], and sequence high-dimensional embeddings and attention maps from RNA-FM [28], where the sequence embeddings and attention maps from RNA-FM are dimensionality-reduced through different MLPs. At each time step of the inverse diffusion process, all these sequence features are fused with the intermediate state x_t. Through this design, RNADiffFold not only effectively harnesses the powerful capabilities of the diffusion model to predict RNA secondary structures but also integrates various sequence features to enhance the accuracy and stability of predictions.

Building upon previous work [19, 21, 22], this study evaluates the predictive performance of RNADiffFold on both within-family and cross-family RNA datasets. Experimental results demonstrate that even with simple one-hot encoding as the conditional input, RNADiffFold exhibits comparable performance to existing methods on within-family datasets,

while also demonstrating reasonable accuracy in predicting the secondary structures of RNA sequences from unknown families. When additional features generated from RNA-FM are incorporated, RNADiffFold shows significant improvements in performance on both within-family and cross-family datasets, surpassing existing methods and highlighting its effectiveness and generalization. Furthermore, it's noteworthy that as a deterministic prediction task of a generative model, RNADiffFold not only predicts static RNA secondary structures but also captures dynamic structural features to some extent by learning the distribution of secondary structure conformations. These dynamic features resemble the pairing and folding of RNA molecules within biological organisms.

Materials and methods

Preliminaries

Diffusion models

Diffusion models [29, 30, 31] are a type of probabilistic generative models characterized by two Markov chains in the diffusion process: a forward chain that deconstructs data into noise, and a reverse chain that reconstructs data from noise. Specifically, in Diffusion Probabilistic Models (DDPMs), given a data distribution $x_0 \sim q(x_0)$, the forward diffusion process q produces a sequence of latent states from x_1 to x_T by adding noise at timestep t with variance schedule $\beta_t \in (0, 1)$. The transition kernel is defined as follows:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathbf{I}) \tag{1}$$

The reverse diffusion process p is parameterized by a prior distribution $x_T \sim \mathcal{N}(0, \mathbf{I})$ and a learnable transition kernel $p_\theta(x_{t-1}|x_t)$ defined as follows:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{2}$$

where θ denotes the model parameters, μ_θ and Σ_θ represent the mean and variance of the distribution at time t. The training objective is to learn the parameters θ so that the reserve trajectory p_θ closely approximates the forward trajectory q. It is achieved by optimizing a variational upper bound on the negative log-likelihood:

$$\begin{aligned}\mathcal{L}_{vb} = \mathbb{E}_{q(x_0)} \Bigg[\sum_{t=2}^{T} \underbrace{D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))}_{\mathcal{L}_{t-1}} \\ + \underbrace{D_{KL}(q(x_T|x_0)||p(x_T))}_{\mathcal{L}_T} - \underbrace{(\mathbb{E}_{q(x_1|x_0)}[\log p_\theta(x_0|x_1)])}_{\mathcal{L}_0} \Bigg]\end{aligned} \tag{3}$$

Due to the incorporation of Gaussian noise as a prior, most diffusion models operate effectively in continuous state spaces; however, they may not efficiently handle discrete data. In addressing this issue, some methods [32, 26, 33] are proposed to generate high-dimension discrete data. For example, D3PM [32] considers a transition matrix with an absorbing state or using discretized, truncated Gaussian distribution. VQ-diffusion [33] accommodates discrete data with a lazy random walk or a random masking operation.

RNA Foundation Model

RNA-FM (RNA Foundation Model) [28] is an RNA foundational model based on the BERT language model architecture, built upon 12 bidirectional encoder blocks based on Transformers. The model consists of two stages: pretraining and fine-tuning. In the pretraining stage, RNA-FM is trained in a self-supervised manner on a large amount of unlabeled RNA sequence data, allowing it to capture latent structural and functional information and extract meaningful RNA representations. Its pretraining strategy is similar to BERT, where 15% of the base tokens representing nucleotides are randomly masked, and the model is trained to reconstruct the masked tokens from the remaining sequence. Upon completion of training, RNA-FM can generate a $640 \times L$ embedding matrix for each RNA sequence of length L. These embedding matrices provide rich feature representations for downstream tasks. In the task-specific fine-tuning stage, the pretrained RNA-FM model can generate sequence embeddings tailored to the requirements of downstream modules, which can be directly used for various RNA-related machine learning tasks. Additionally, if higher task performance is needed, the model can be further fine-tuned by adding lightweight prediction layers to adapt to more specific task scenarios.

It is worth noting that a recent study [34] further validated the importance of the multi-head attention mechanism outputs in RNA-FM for capturing structural information. These attention maps not only reveal the strength of associations between different positions in RNA sequences but also provide new perspectives for understanding RNA secondary structure and function.

Architecture of RNADiffFold

In this part, we will introduce the diffusion process and the condition construction strategy. Before delving into the details of RNADiffFold, we outline its overall pipeline. As depicted in Figure 2A. In the left branch, the input RNA sequence undergoes a conditional construction unit to obtain four types of feature representations: one-hot encoding, probability maps, embeddings from RNA-FM, and attention maps from RNA-FM. In the right branch, the RNA secondary structure is represented as an $L \times L$ binary contact map. During the diffusion process, discrete noise is gradually injected to disrupt the original contact map, and after T time steps, the contact map transitions into completely random noise. In the reverse diffusion process, denoising is performed using a U-Net denoising network, combined with sequence features outputted by the conditional control unit, to progressively restore the original contact map. Once the model training is completed, given a randomly sampled noise x_T and an RNA sequence, the progressive denoising process can predict the secondary structure contact map. Below, we will introduce the diffusion model component and the conditional construction unit separately.

Diffusion process.

As illustrated in Figure 2C, RNADiffFold employs a shared-weight neural network to learn the progressive reconstruction of data over T steps. The diffusion process of RNADiffFold is implemented based on multinomial diffusion [26], but there are some differences. Specifically, the model deals with binary contact maps where pixel values are limited to two representations: 0 and 1. The initial x_0 is a $L \times L$ tensor with deterministic 0-1 relationships, where L represents the sequence length. To undergo denoising learning via U-Net, each pixel is embedded into an 8-dimensional vector, resulting in a $L \times L \times 8$ tensor representation. Subsequently, using the Gumbel-Softmax method [35], discrete noise is gradually added to the sample at

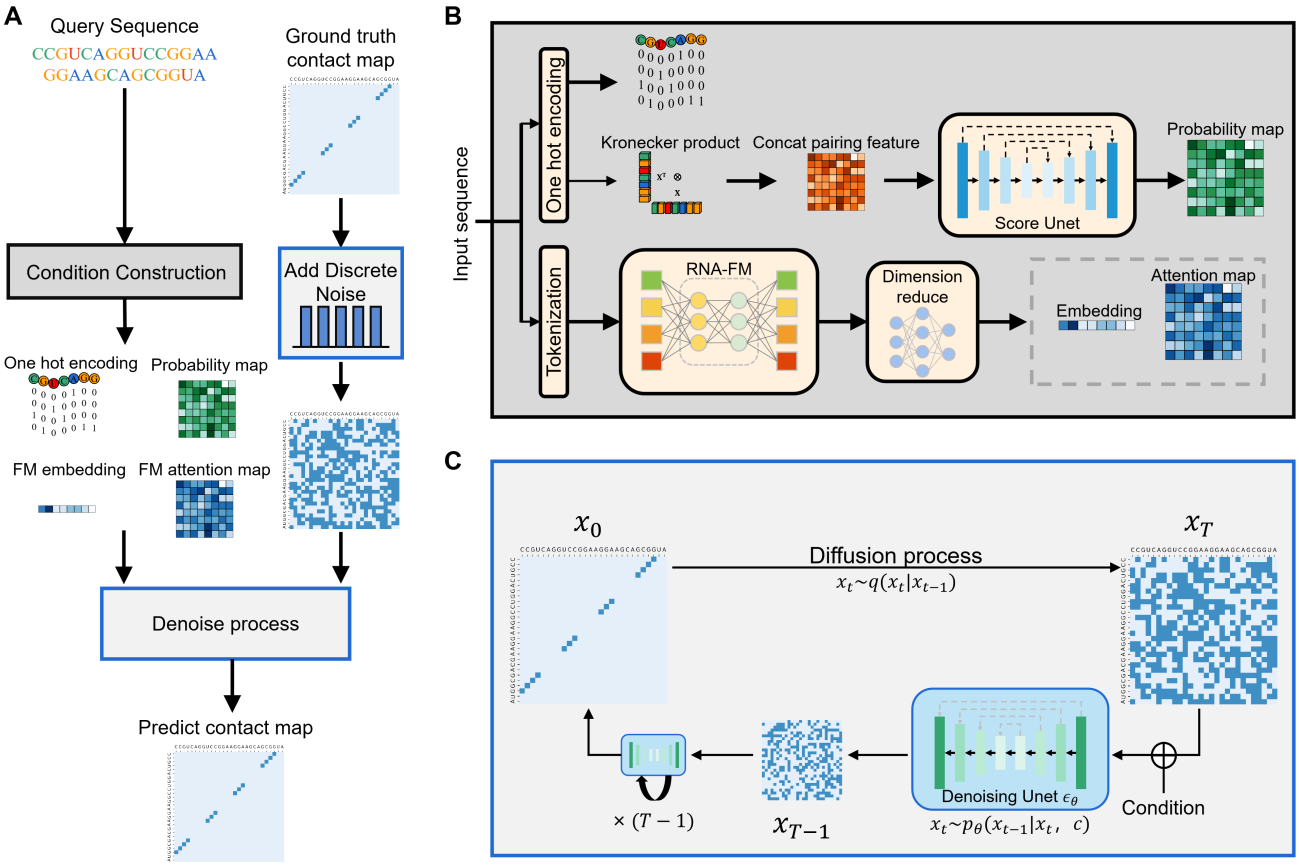

Fig. 2. The Pipeline of RNADiffFold. (**A**)Overview of RNADiffFold workflow as a supervised task. Given the query sequence, we construct the representation through the condition construction stage. During the training stage, discrete noise is incrementally added to the ground truth contact map. Then, we gradually denoise the map conditioned by the sequence representation. In the prediction stage, given the sequence and a map randomly sampled from the categorical distribution, we generate candidate maps with different seeds and vote for the most reasonable one. (**B**) Details of condition construction. We leverage RNA-FM and UFold score networks with their respective pre-trained weights. Following the corresponding operations, we get four condition representations: one hot encoding, probability map, FM embedding, and FM attention map. (**C**)Details of the diffusion process. We leverage multinomial diffusion and a Unet network as the denoising network for training and inferring.

each timestep t through the forward process defined as follows:

$$q(x_t|x_{t-1}) = \text{Multinomial}((1-\beta_t)\cdot x_{t-1} + \beta_t \cdot \frac{1}{K}\cdot \vec{\mathbf{1}}) \quad (4)$$

where x_t represents a contact matrix at time t in one-hot encoded format $x_t^{ij} \in \{0,1\}^K (K = 2, 0 \le i,j \le L)$. $\vec{\mathbf{1}}$ is an all-one matrix. β_t is the chance of resampling another pairing possibility uniformly. We apply the cosine schedule [31] to avoid spending many steps on high-noise problems. As t approaches T, β_t is adjusted to approximate 1, making the distribution closer to the uniform distribution. Since it is Markovian, we can sample arbitrary x_t directly based on x_0 as:

$$q(x_t|x_0) = \text{Multinomial}(\bar{\alpha}_t \cdot x_0 + (1-\bar{\alpha}_t)\cdot \frac{1}{K}\cdot \vec{\mathbf{1}}) \quad (5)$$

where $\alpha_t = 1-\beta_t$, and $\bar{\alpha}_t = \prod_{\tau=1}^{t} \alpha_\tau$. Using Eq. 4 and Eq. 5 we can derive the categorical posterior $q(x_{t-1}|x_t, x_0)$ in the following form:

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \text{Multinomial}\left(\tilde{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{x}_0) / \sum_{k=1}^{K} \tilde{\theta}_k\right) \quad (6)$$

where $\tilde{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{x}_0) = [\alpha_t \cdot \boldsymbol{x}_t + \frac{(1-\alpha_t)}{K}] \odot [\bar{\alpha}_{t-1} \cdot \boldsymbol{x}_0 + \frac{(1-\bar{\alpha}_{t-1})}{K}]$. The generative diffusion process is defined as:

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \hat{\boldsymbol{x}}_0) = \text{Multinomial}\left(\tilde{\boldsymbol{\theta}}(\boldsymbol{x}_t, \hat{\boldsymbol{x}}_0) / \sum_{k=1}^{K} \tilde{\theta}_k\right) \quad (7)$$

where $\hat{x}_0 = \mu(x_t, t, c)$ is a neural network to predict the initial contact matrix $\hat{x}_0$ given the previous step state x_t and condition $c \in \{c_{onehot}, c_u, c_{emb}, c_{attn}\}$. Note that, we parametrize $p(x_{t-1}|x_t)$ using the probability vector from $q(x_{t-1}|x_t, \hat{x}_0)$. The main difference between these two processes lies in the fact that the forward diffusion process introduces to the data, making it independent of data or condition. Conversely, the generative diffusion process relies on the provided condition and a comprehensive observation of the preceding step.

The training objective of this diffusion process is to minimize the expected Kullback–Leibler(KL) divergence between Eq. 7 and Eq. 6, following a similar form as in Eq. 3:

$$\begin{aligned} \mathcal{L}_{t-1} &= D_{KL}(q(x_{t-1}|x_t, x_0)||p(x_{t-1}|x_t)) \\ &= D_{KL}(\mathcal{C}(\theta_{post}(x_t, x_0))||\mathcal{C}(\theta_{post}(x_t, \hat{x}_0))) \end{aligned} \quad (8)$$

Furthermore, $\mathcal{L}_T$ corresponds to the second term in Eq 3. Given that x_0 is one-hot encoded, the computation of $\mathcal{L}_0$ can be

expressed as:

$$\mathcal{L}_0 = \log p_\theta(x_0|x_1) = \sum_k \boldsymbol{x}_{0,k} \log \hat{x}_{0,k} \tag{9}$$

Details of denoising the Unet network are illustrated in Supplementary Figure S1. Benefiting from the flexibility of Unet to handle inputs of different sizes, we can also process variable-length sequence data.

Condition construction unit.
The condition construction is depicted in Figure 2B. To incorporate features of the input sequence into the reverse diffusion process, the most intuitive strategy is to perform one-hot encoding on the sequence. Experimental results demonstrate that when using one-hot encoded features c_{onehot} as conditions, RNADiffFold exhibits a certain level of secondary structure prediction capability, comparable to state-of-the-art methods on some datasets. However, its predictive performance decreases when facing more complex scenarios. Therefore, it is necessary to construct additional neural networks to process input sequences, thereby extracting more meaningful features and generating sequence representations containing rich additional information.

As illustrated in Figure 2B, we adopt two neural networks as feature extractors, namely the Ufold scoring network and the pre-trained RNA-FM model. Here, the conditional representations from Ufold consist of a probability matrix c_u, while those from RNA-FM consist of a sequence embedding c_{emb}, and an attention map c_{attn}. Given an RNA sequence of length L, denoted as $\mathbf{s} = (s_1, s_2, \ldots, s_L)$, where each $s_i \in \{A, U, C, G, N\}$ and N represents an unknown state, the computation methods for the four types of feature representations are described as follows: c_{onehot} is the sequence feature from one-hot encoding, $c_{onehot} \in \{0,1\}^{4\times L}$. The encoding rules are as follows: A: (1, 0, 0, 0), U: (0, 1, 0, 0), C: (0, 0, 1, 0), G: (0, 0, 0, 1), N: (0, 0, 0, 0). c_u is the probability matrix outputted by the scoring network from Ufold. Specifically, firstly, the Kronecker product is computed between c_{onehot} and itself, and then the dimensions are adjusted to transform c_{onehot} into a tensor $c_{kronecker} \in \{0,1\}^{16\times L\times L}$. Subsequently, to address the sparsity issue in class-like representations, $c_{kronecker}$ is concatenated with an additional pairing probability matrix used in CDPFold [36], resulting in a tensor c_{input} of size $17 \times L \times L$. Thus, the obtained feature representation considers all potential pairing possibilities without imposing explicit constraints, thereby enabling the prediction of more complex structures. Finally, c_{input} is inputted into a U-Net network to produce the probability tensor c_u, with dimensions of $8 \times L \times L$. In this process, to meet the dimension requirements of the diffusion process, the last layer of the U-Net network is adjusted so that its output dimension is exactly 8. It is noteworthy that apart from the last layer, the initial weights of the U-Net are sourced from the pre-trained Ufold model. Subsequently, fine-tuning operations are conducted to ensure the entire network performs optimally for the current task.

c_{emb} and c_{attn} are respectively the one-dimensional sequence embedding and the two-dimensional attention map from RNA-FM. Specifically, the RNA sequence is inputted into the pre-trained RNA-FM model. From each encoder block's Multi-Head Attention (MHA) layer, attention maps of dimensions $20 \times L \times L$ are obtained, and from the output layer, the sequence embedding c_{emb} of dimensions $640 \times L$ are derived, where 640 represents the embedding dimension, and 20 represents the number of attention heads. Subsequently, the attention maps from the 12 encoder blocks are integrated to form the final attention map c_{attn} of dimensions $240 \times L \times L$. c_{emb} contains meaningful biological information. As argued in the [34], the multiple attention heads in different layers of the Transformer encoder module should theoretically capture structural information by focusing on the strength of correlations between different positions in the input sequence. Therefore, c_{attn} contains information related to the structure.

Datasets preparation

Referring to prior work [19, 21, 22], we employed several datasets to evaluate the performance of RNADiffFold in both within-family and cross-family scenarios. These datasets include RNAStrAlign [37], ArchiveII [38], bpRNA-1m [39], bpRNA-new [20], and PDB. A summary of the datasets utilized in our experiments can be found in Supplementary Table S1. Specifically, the RNAStrAlign, ArchiveII, and bpRNA-1m datasets were utilized for evaluating within-family performance, while the bpRNA-new and PDB datasets were employed for assessing cross-family performance.

RNAStrAlign comprises 30,451 unique sequences distributed among 8 RNA families, whereas ArchiveII consists of 3,975 sequences spanning 10 RNA families. These datasets serve as widely used test sets, with RNAStrAlign commonly used as the training set due to their similar family distribution. The bpRNA-1m dataset contains 102,318 sequences from 2,588 families and is considered one of the most exhaustive datasets used for benchmarking RNA secondary structure prediction methods. On the other hand, the bpRNA-new dataset was derived from Rfam14.2 [40] and includes 1,500 new RNA families detected by newly developed techniques [41]. Furthermore, the PDB dataset, collected from the Protein Data Bank [42], comprises high-resolution RNA structures with resolutions less than 3.5$\mathring{A}$ (as of April 9, 2020).

Before using these datasets, preprocessing steps were conducted. Specifically, for the RNAStrAlign dataset, sequences longer than 640 nucleotides were first selected for computational efficiency. Subsequently, the dataset was randomly split into training, validation, and test sets in an 8 : 1 : 1 ratio. Finally, the same processing method as E2Efold [18] and Ufold [19] was applied to remove redundant sequences among these sets. For the ArchiveII dataset, which was used solely for testing, a similar strategy as the RNAStrAlign dataset was adopted to reduce redundancy. Regarding the bpRNA-1m dataset, sequences with a similarity exceeding 80% were removed using CD-HIT-EST [43], following the methods of MXfold2 [20] and SPOT-RNA [21]. The remaining data, referred to as bpRNA, was randomly divided into training, validation, and test sets labeled as TR0, VL0, and TS0, respectively. For the bpRNA-new dataset, sequences with a similarity exceeding 80% were also removed using CD-HIT-EST, and sequences longer than 500 nucleotides were filtered out. For the PDB dataset, the same dataset division as SPOT-RNA2 [22] was applied, with TR1 for training, VL1 for validation, and TS1, TS2, TS3, and TS-hard for testing.

Furthermore, we analyzed the relationship between sequence length and the number of base pairs in each dataset. As shown in Supplementary Figure S2, the results revealed a linear relationship between sequence length and the number of base pairs in most dataset samples. However, some samples deviated from the fitted line, with most of the deviations related to sequences containing low pairing information and some being completely unpaired. To mitigate the impact of these data points, sequences longer than 200 nucleotides and with fewer than 10

base pairs were removed from all samples used for training. The validation and test sets remained consistent with previous work. Therefore, the final dataset used in this study was slightly smaller than that used in previous studies.

To enhance the transfer ability of the model in handling unknown RNA families, we employed the same data augmentation strategy as Ufold [19]. Firstly, in the bpRNA-new dataset, we randomly mutated $20\% - 30\%$ of the nucleotides to generate new data. Then, sequences with a similarity greater than 80% to real sequences were removed using CD-HIT-EST. Finally, 2717 mutated sequences were obtained for supplementary training data, named Mutate-data. The labels for Mutate-data were generated by Contrafold [44], a probabilistic method that extends stochastic context-free grammars (SCFGs) by using discriminative training objectives and flexible feature representations.

In summary, the training data for this study consisted of several datasets' training sets, including those from RNAStrAlign, TR0, and Mutate-data, as well as TR1 for fine-tuning. The test datasets comprised ArchiveII and TS0 for within-family testing, and bpRNA-new and the test set from PDB (TS1, TS2, TS3, and TS-hard) for cross-family testing.

Previous approaches typically padded all training data to the maximum sequence length for batch training, which resulted in lower training efficiency. To improve training efficiency, we implemented a data bucketing strategy to handle the datasets described above. Specifically, for each dataset, sequences were sorted by length and then binned with intervals of 80 nucleotides, ensuring that sequences within the same bucket had similar lengths. Subsequently, sequences within each bucket were padded to the maximum length of that bucket. For example, sequences with lengths between 80 to 160 nucleotides were grouped into the same bucket and uniformly padded to 160 nucleotides. This approach ensured that shorter sequences were not padded to excessively large lengths. During training, buckets were utilized for training, and the batch size was set to 1. This bucketing strategy alleviated unnecessary computations in padding regions, ensuring efficient data processing and reducing training time by approximately 50%.

Additionally, it was observed that approximately 80% of the sample sequences in the original data were concentrated below 160 nucleotides in length (23,374 sequences between 30 and 160 nucleotides, 3,202 sequences between 160 and 320 nucleotides, and 3,119 sequences between 320 and 640 nucleotides), as shown in Figure S3(A), which hindered the model's ability to learn long-range RNA interactions. To address this issue, a data balancing strategy [45] was employed. Specifically, sequences with lengths between 160 and 640 nucleotides were selectively duplicated in the training set to increase their representation in the dataset. After duplication, the number of sequences with lengths between 160 and 640 nucleotides increased to 48658. This initiative effectively mitigated the impact of data imbalance on model training. Figure S3(B) shows the distribution of sequence length and number of base pairs after data balance.

Results

Evaluation criteria and experimental setup

To comprehensively evaluate the prediction accuracy of RNADiffFold, we employed precision (*Prec*), recall (*Recall*), and F1 score as evaluation metrics, defined as follows:

$$Prec = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1 = 2 \times \frac{Prec \times Recall}{Prec + Recall} \tag{12}$$

where TP represents the number of correctly predicted base pairs (true positives), FP represents the number of incorrectly predicted base pairs (false positives), and FN represents the number of base pairs in the reference structure that were not predicted (false negatives).

RNADiffFold is implemented using Pytorch[46]. During the diffusion process, we set the number of diffusion steps T as 20 for the contact map containing sparse classification information. Large T, on the contrary, tends to degrade prediction performance. Adam[47] is employed as the optimizer with a 1e-3 learning rate. The model undergoes training for a maximum of 400 epochs, and validation is performed every 20 epochs. Early stopping with a patience of 5 is implemented to prevent overfitting. We divided the training process into two stages. Firstly, the model was trained on the RNAStrAlign training set, TR0, and Mutate-data, followed by testing on ArchiveII, TS0, and bpRNA-new to evaluate its generalization ability. Secondly, using the model weights obtained in the first stage, fine-tuning was performed on the PDB dataset, and testing was conducted on TS1, TS2, TS3, and the challenging TS-hard dataset to validate the model's performance in different scenarios. All experiments are carried out on a single Nvidia A40 GPU and the CPU version of Intel(R) Xeon(R) Gold 6326 CPU @ 2.90GHz.

Baselines

We compared RNADiffFold with 10 competitive baseline methods for RNA secondary structure prediction. These methods include six energy-based approaches: Mfold [48], Linearfold [13], RNAfold [49], RNAStructure [11], Contrafold [44], and ContextFold [50], as well as four recently developed deep learning-based methods: SPOT-RNA [21], MXfold2 [20], E2Efold [18], and UFold [19]. In the energy-based methods, Mfold utilizes nearest-neighbor energy parameters and dynamic programming to find the structure with the lowest energy. Linearfold combines dynamic programming with beam search to enhance efficiency. RNAfold integrates dynamic programming algorithms with a thermodynamic-based energy model to predict optimal RNA structures. RNAStructure calculates minimum free energy and optimizes prediction results based on experimental data. Contrafold uses conditional log-linear models, extending to stochastic context-free grammars (SCFGs), and integrates thermodynamic parameters. ContextFold introduces a fine-grained model with a large parameter set (approximately 70,000). Among the deep learning-based methods, SPOT-RNA employs a CNN and BiLSTM architecture and utilizes transfer learning to enhance performance. MXFold2 combines deep learning with thermodynamic parameters for more accurate predictions. E2Efold innovatively integrates the Transformer model with an unrolling algorithm, imposing hard constraints on RNA secondary structure. UFold utilizes U-Net and a structure similar to image inputs to effectively capture long-range dependencies. It is worth noting that due to the lack of reproducible results from SPOT-RNA and ContextFold in experiments, we referenced the experimental data of UFold [19] for comparison analysis.

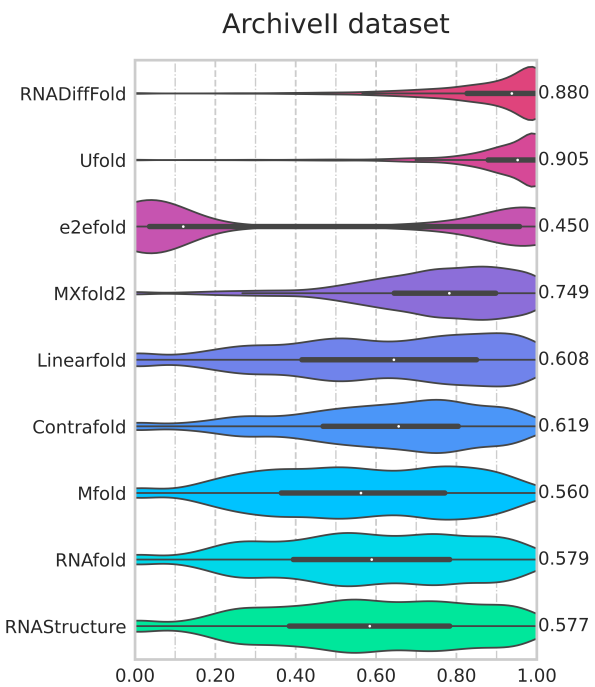

Fig. 3. Violin plot on the ArchiveII dataset. Visualization of F1 value of RNADiffFold against 8 other methods.

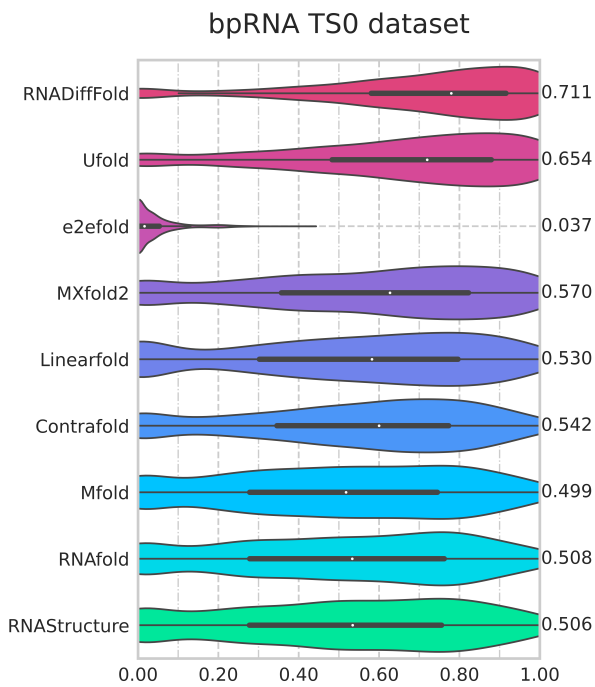

Fig. 4. Violin plot on the bpRNA TS0 dataset. Visualization of F1 value of RNADiffFold against 8 other methods.

Performance on within-family datasets

The "within-family" datasets indicate that the RNA families in the test set are highly similar to those in the training set. As reported in previous studies, the evaluation results on these datasets effectively reflect the predictive performance of the model. The analysis of these datasets shows mixed results. As shown in Figure 3 and Supplementary Table S2, for the ArchiveII test set, RNADiffFold demonstrates good predictive performance with an F1 score of 0.880. This result not only surpasses all energy-based methods but also ranks among the top in deep learning-based methods. Its F1 score is slightly lower than that of UFold and RNA-FM, which may be attributed to information loss caused by the conditional construction strategy. Since ArchiveII contains a limited number of families and species, RNADiffFold achieves comparable predictive performance using only one-hot encoding as a condition, as further confirmed in subsequent ablation experiments.

In contrast, the bpRNA dataset from Rfam12.2 [51] encompasses a broader range of RNA families. The evaluation results on its test set, TS0, are depicted in Figure 4 and Supplementary Table S2, where the results for SPOT-RNA and ContextFold are taken from the original UFold paper. RNADiffFold achieves an average F1 score of 0.711, surpassing all energy-based and deep learning-based methods. Specifically, RNADiffFold shows a notable 8.7% improvement in F1 score compared to UFold and a 2.7% increase compared to RNA-FM. This significant performance enhancement not only highlights the superiority of RNADiffFold in the field of RNA secondary structure prediction but also demonstrates that its outstanding performance stems not merely from the simple combination of the UFold scoring network and RNA-FM output features but rather from its unique algorithm design and feature fusion strategy.

To comprehensively assess RNADiffFold's capability in predicting long-range base pairs, we followed the methodology of UFold [19] and conducted an in-depth analysis on the bpRNA TS0 dataset. Specifically, for each RNA sequence of length L, we classified base pairs and non-base pairs with intervals exceeding $L/2$ as long-range base pairs. After rigorous screening, we selected 993 samples from a total of 1,304 samples that met the criteria for subsequent precision, recall, and F1 score calculations. We compared RNADiffFold with other mainstream methods, and the results are presented in Table 1 and Figure S4. These numerical results demonstrate RNADiffFold's outstanding performance on long-range base pair RNA data. Compared to UFold, RNADiffFold's predicted precision and recall are closer, indicating its more stable predictive performance. This stability may stem from the effective integration of the pre-trained UFold scoring network with RNA-FM output features, providing RNADiffFold with rich contextual information and structural features, thereby enhancing its accuracy in predicting long-range base pairs.

Table 1. F1 scores of RNADiffFold compared with other learning-based methods on the TS0 dataset of long range base pairing.

Method	Precision	Recall	F1
RNADiffFold	**0.748**	0.764	**0.739**
Ufold	0.644	**0.765**	0.687
SPOT-RNA[a]	0.361	0.492	0.403
MXfold2	0.318	0.450	0.360
e2efold	0.038	0.084	0.043

[a] The results of SPOT-RNA are cited from Ufold[19]

Performance on cross-family datasets

The "cross-family" datasets indicate that there is no similarity between the RNA family species in the test set and those in the training set. Therefore, the performance on such datasets, particularly for deep learning-based methods, can better reflect their understanding of RNA folding patterns and their ability

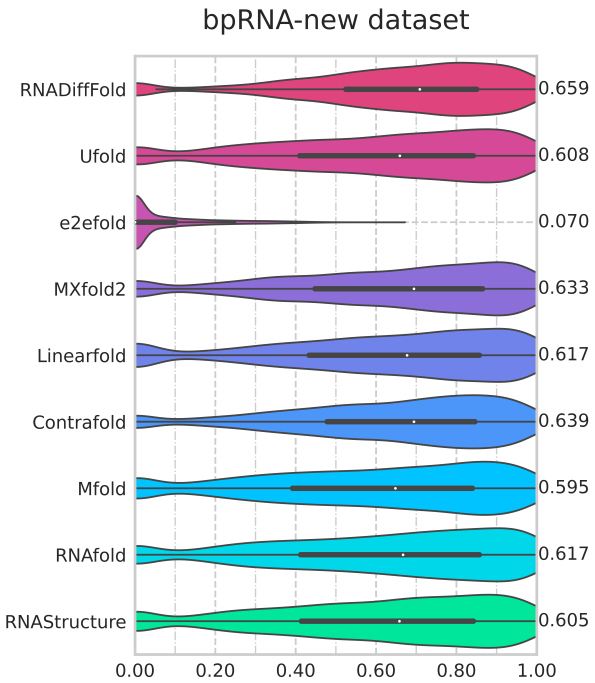

Fig. 5. Violin plot on the bpRNA-new dataset. Visualization of F1 value of RNADiffFold against 8 other methods.

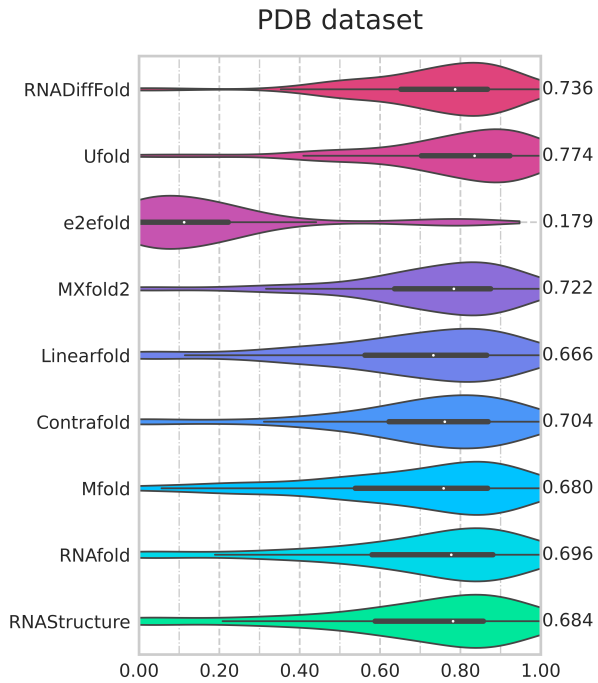

Fig. 6. Violin plot on the PDB test dataset encompassing TS1, TS2 and TS3. Visualization of F1 value of RNADiffFold against 8 other methods.

to generalize knowledge to unknown families. To enhance the model's generalization performance, we adopted UFold's data augmentation strategy, generating 2717 mutated sequences with mutation rates ranging from 20% to 30% from the original sequences. These sequences were then predicted for their secondary structures by Contrafold to serve as pseudo-labels. Subsequently, these mutated datasets were integrated into RNAStrAlign and TR0 for training and comprehensively evaluated on three different test sets. As shown in Figure 5 and Supplementary Table S3, RNADiffFold outperformed other methods, demonstrating its superior performance in predicting the secondary structures of RNA sequences from unknown families. Unlike baseline methods like MXfold2, which incorporate thermodynamic constraints, or UFold, which employs biological-based hard constraints, RNADiffFold further predicts the secondary structures of new unknown family sequences by learning the distribution of RNA conformations. Additionally, we further analyzed the performance of different length data by partitioning the bpRNA-new dataset at intervals of 100 nt. As shown in Figure S5, the performance of most methods deteriorates with increasing length. RNADiffFold maintains excellent predictive performance for sequences below 300 nt but decreases more for sequences between 300 and 500 nt. This may also be attributed to the fact that RNADiffFold does not include any constraints.

To comprehensively evaluate the predictive performance of RNADiffFold in cross-family scenarios, we conducted additional validation using the PDB dataset. The PDB dataset [42] contains secondary structure information extracted from high-resolution RNA 3D structures, providing a reliable basis for model evaluation. Following the same dataset partitioning method as SPOT-RNA2 [22], we fine-tuned the pre-trained model on the PDB dataset. Figure 6 illustrates the average F1 score of RNADiffFold across the entire PDB dataset, reaching 0.736, demonstrating comparable performance with other methods. To further analyze the performance of RNADiffFold on different test sets within PDB, we evaluated it on TS1, TS2, and TS3 separately. Table S4 and Figure S6 provide detailed performance metrics of RNADiffFold on these three test sets. Additionally, we constructed a more challenging test set, TS-hard, which was curated from TS1 and TS3 by excluding similar sequences using BLAST-N [52] and INFERNAL [53] models. The evaluation results on TS-hard, as shown in Table S5 and Figure S7, validate that RNADiffFold maintains comparable performance with most methods even in challenging prediction tasks.

To further substantiate the predictive capability of RNADiffFold, we employed two statistical significance assessment methods: individual t-test-based method and bootstrapping-based method. The results in Table S6 indicate that RNADiffFold significantly outperforms other methods statistically, with most p-values being less than 0.05. For the relatively small-scale PDB dataset, bootstrapping proved to be a more effective evaluation method. Table S7 and Figure S8 present the 95% confidence intervals of F1 scores for all compared methods on the PDB test sets (TS1, TS2, TS3), providing robust support for the earlier conclusions.

Multiple sampling and voting strategy testing experiments

To further enhance the performance of the model, this study implemented a strategy based on multiple sampling and voting. Given that RNADiffFold is a generative prediction method with the ability to learn the distribution of secondary structures, we utilized random seeds to generate a set of predicted structures from the model. Subsequently, we conducted cluster analysis by calculating the similarity between these predicted structures, and selected a representative structure from the largest cluster as the final prediction. In cases where there were no clear clusters, a random selection method was employed to determine the final

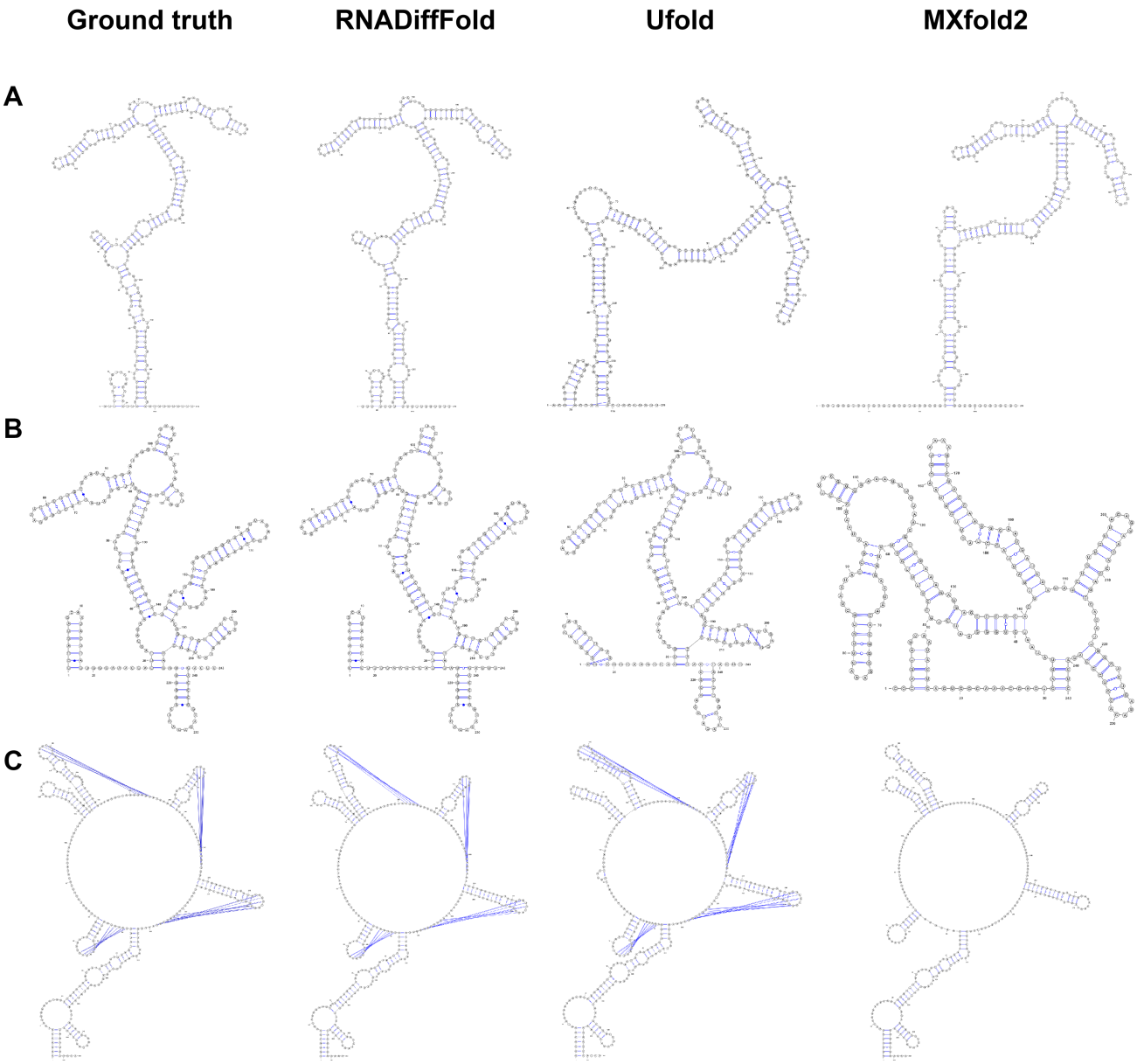

Fig. 7. Visualization comparing three examples predicted by RNADiffFold with those from two other methods against the ground truth. The RNA sequences are from the following families: (A) Aspergillus fumigatus, recorded in the SRPDB database; (B) Alphaproteobacteria subfamily 16S rRNA, recorded in the RNAStrAlign database; and (C) Escherichia coli, recorded in the tmRNA database. The results indicate that RNADiffFold aligns more closely with the ground truth in details than other two methods.

prediction. Application of this strategy resulted in a certain degree of performance improvement, as shown in Table S8. Specifically, by increasing the number of samples from 1 to 10, moderate improvements were observed in the evaluation metrics across the four test sets. This outcome suggests that the strategy based on multiple sampling and voting helps to reduce the uncertainty of model predictions, thereby enhancing the accuracy of RNA secondary structure prediction.

Visualization

To intuitively demonstrate the model's ability to capture the details of RNA secondary structures, we randomly selected RNA sequences from different RNA family species and visually compared the predicted structures from RNADiffFold with those from two other top-performing methods, Ufold and MXfold2, as well as the ground truth. Specifically, the predicted results were converted into ct format based on base pairing relationships and then visualized using VARNA[54]. Figure 7 shows that the structures predicted by RNADiffFold are closer to the ground truth than those predicted by other methods, demonstrating improved accuracy in detail.

We also validated RNADiffFold's ability to capture dynamic structural features using CoDNaS-RNA[55], a database containing diverse RNA conformational ensembles. Considering data quality and sequence length, we selected 14 clusters, each containing two different conformations of the same sequence. Among these, 9 clusters had sequences that were extremely

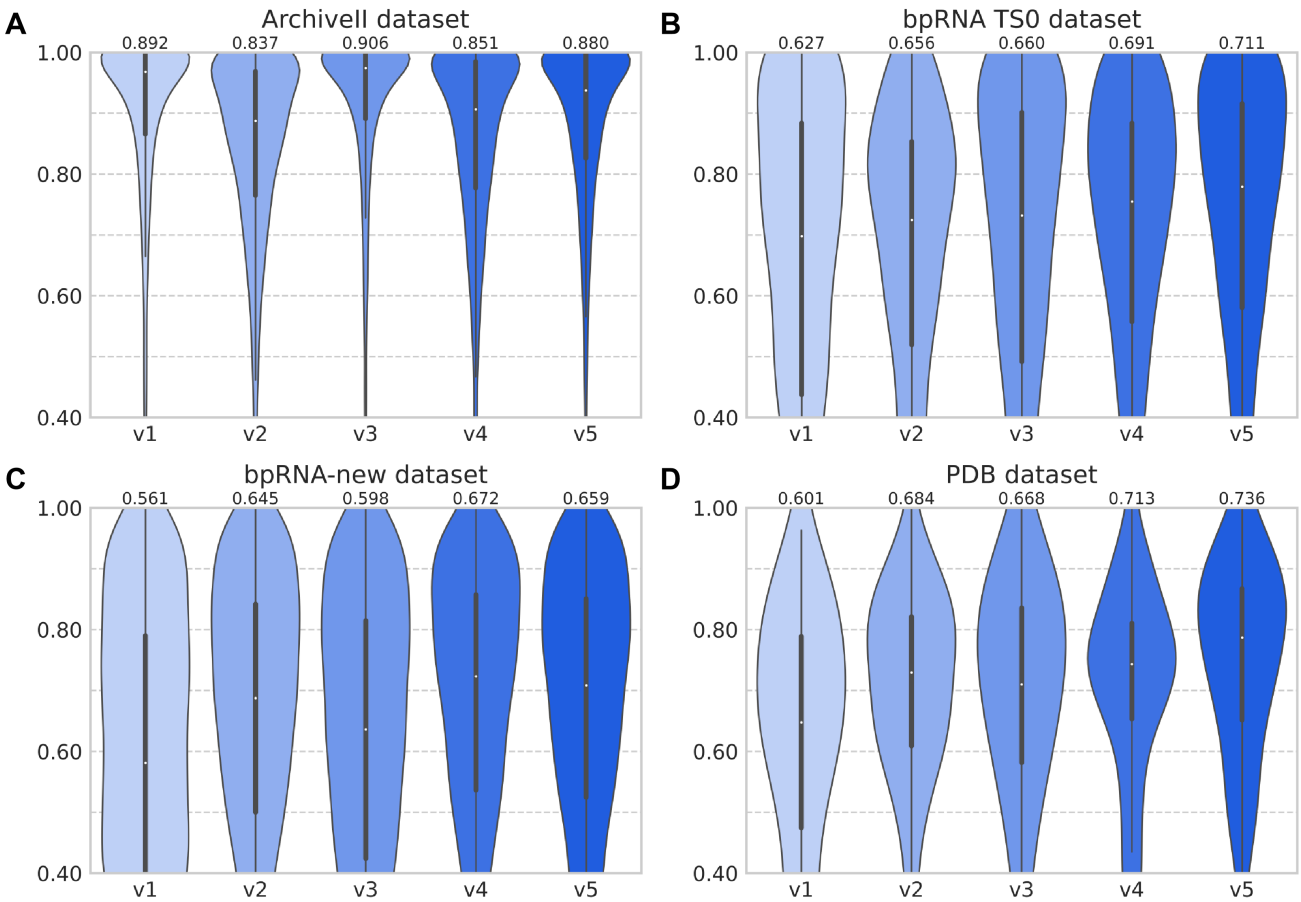

Fig. 8. F1 values on the four datasets(within-family datasets ArchiveII(A), bpRNA TS0(B) and cross-family datasets bpRNA-new(C), PDB datasets). v1-v5 are different condition construction strategies combined with diffusion process. v1: one-hot encoding c_{onehot}; v2: probability map from the Ufold score network c_u; v3: one-hot encoding c_{onehot}, output from RNA-FM c_{emb}, c_{attn}; v4: one-hot encoding c_{onehot}, probability map from the Ufold score network c_u, output from RNA-FM c_{emb}; v5: full version of RNADiffFold.

similar, differing by only 1-2 bases. The experimental results indicate that RNADiffFold successfully predicted the structural profiles of 11 clusters and captured the different conformations to a certain extent. Figure S10 shows the examples of 4 clusters, with Cluster 189 being representative of the predictions for the 9 similar clusters. As for the failed cases, these occurred because the ground truth sequences tended to be completely unpaired or contained pairings that violated standard pairing rules. Consequently, the model was unable to make accurate predictions, as shown in Figure S11.

Ablation Study

In this section, we first discuss the impact of different conditioning strategies on the performance of RNADiffFold while keeping other experimental parameters constant. The versions with different conditions are labeled as follows: v1: using only the one-hot encoding c_{onehot} of the given sequence; v2: utilizing the probability map c_u from the Ufold scoring network; v3: employing both c_{onehot} and c_{emb} and c_{attn} from RNA-FM simultaneously; v4: utilizing c_{onehot}, c_u, and c_{emb} as conditions simultaneously; v5: RNADiffFold's final form, incorporating all four conditions simultaneously.

Figure 8 along with Table S9 comprehensively present the performance evaluation results of RNADiffFold on four test sets. Key observations drawn from this study are as follows: (1) On the ArchiveII test set and TS0, RNADiffFold exhibits competitiveness with state-of-the-art methods (Ufold and RNA-FM) when using only c_{onehot} as a condition (i.e., version v1). However, the performance of the v1 version appears relatively limited when faced with sequences from unknown families, indicating the inadequacy of one-hot encoding in feature extraction. (2) The probability map c_u from the Ufold scoring network (corresponding to version v2) demonstrates a greater contribution on cross-family data, while c_{emb} and c_{attn} from RNA-FM (corresponding to version v3) play a more significant role in within-family data. Therefore, by integrating these two sets of output features as sequence condition information, more balanced performance can be achieved on both within-family and cross-family data. (3) By comparing versions v4 and v5, the utility of the attention map c_{attn} can be understood. After introducing c_{attn}, the model shows slight performance improvements on almost all datasets, albeit modest in magnitude. This may be attributed to partial feature information loss due to excessive dimensionality reduction.

Additionally, in the early experiments, we also explored the relationship between different diffusion step sizes and model performance and unexpectedly found that smaller step sizes perform better in the diffusion process, as shown in Figure S9. This may be because the secondary structure contact map

contains sparser information compared to images from the real world. Conversely, larger step sizes may make it difficult for the model to explore the conformation distribution space. These findings provide valuable guidance for further optimizing RNADiffFold.

Discussion

In this paper, we present RNADiffFold, a novel RNA secondary structure prediction method based on a discrete diffusion model. Unlike traditional deep learning-based prediction methods, RNADiffFold treats secondary structure prediction as a pixel-level contact map segmentation task and utilizes discrete diffusion processes to capture conformational distributions. In the forward diffusion process, noise following a uniform distribution is gradually injected into the true contact map until it is completely randomized. In the reverse diffusion process, RNADiffFold employs UNet as a denoising network. Additionally, we propose an effective conditioning strategy to extract condition information from RNA sequences to guide structure prediction.

Compared to previous works, RNADiffFold offers significant advantages. Firstly, prior methods typically predict secondary structures in a deterministic manner, which contradicts the dynamic folding nature of RNA. However, RNADiffFold explores the conformational space of RNA through diffusion processes without imposing any explicit hard constraints, enabling it to predict non-canonical pairings arising from tertiary interactions. Secondly, our proposed conditioning strategy maximizes the advantages of different models. From another perspective, RNADiffFold can be viewed as a downstream task of pre-trained models, avoiding the embedding of complex prior knowledge. Experimental results demonstrate that RNADiffFold achieves competitive performance in predicting RNA secondary structures compared to current deep learning and energy-based methods. Furthermore, the method exhibits the ability to capture dynamic RNA features. Further experiments on intra-family and inter-family datasets validate the effectiveness and robustness of RNADiffFold.

Although RNADiffFold demonstrates outstanding predictive performance, there is still room for improvement. Firstly, a more specific design of the loss function for training the diffusion model is needed to address contact map features containing sparse information. Unlike real-world images, contact maps contain fewer category information, with most regions being classified as "0" class. Therefore, resources should be reduced in these easily learnable regions. Secondly, considering adopting more efficient conditional pre-training models to enhance performance. Currently, due to computational cost limitations, this study has not retrained a condition control model tailored for the diffusion model. In theory, using more suitable and thoroughly trained pre-training models to extract sequence features as conditions is expected to further improve the performance of RNADiffFold.

Key Points

- We introduce RNADiffFold, a novel discrete diffusion framework that treats RNA secondary structure prediction as a pixel-level segmentation task. This approach effectively captures the distribution of secondary conformations, unveiling valuable biological insights embedded in the primary sequence.
- A condition construction strategy is proposed to construct sequence features, enabling flexible feature design to enhance prediction performance or cater to specific tasks.
- Experiments show that RNADiffFold surpasses previous energy-based and recent learning-based methods on within- and cross-family datasets, demonstrating the effectiveness and robustness of RNADiffFold.

Data and Code Availability

The code to reproduce our experiments and source data is available at `https://github.com/HIM-AIM/RNADiffFold` under an MIT License.

Author Contribution

Y.F., Z.W., P.Y., and X.L. conceived the research project. X.L., and Y.P. supervised and advised the research project. Y.F., and Z.W. designed and implemented the RNADiffFold framework. Y.F., Z.W., Q.T., and Z.L. conducted the computational analyses. Y.F., Z.W., and X.L. wrote the manuscript. All the authors discussed the experimental results and commented on the manuscript.

Acknowledgments

We acknowledge helpful discussions with members of the AIM lab. The authors thank the anonymous reviewers for their valuable suggestions.

Funding

This work is supported in part by funds from the National Key Research and Development Program of China (2022YFC3600902).

Competing Interests Statement

The authors declare no competing interests.

References

1. Matthew G Seetin and David H Mathews. Rna structure prediction: an overview of methods. *Bacterial regulatory RNA: methods and protocols*, pages 99–122, 2012.
2. Scott M. Hammond. An overview of micrornas. *Advanced Drug Delivery Reviews*, 87:3–14, Jun 2015.
3. Tim R. Mercer, Marcel E. Dinger, and John S. Mattick. Long non-coding rnas: insights into functions. *Nature Reviews Genetics*, 10(3):155–159, Mar 2009.

4. Tomaž Bratkovič, Janja Božič, and Boris Rogelj. Functional diversity of small nucleolar rnas. *Nucleic Acids Research*, 48(4):1627–1651, Feb 2020.
5. Jörg Fallmann, Sebastian Will, Jan Engelhardt, Björn Grüning, Rolf Backofen, and Peter F. Stadler. Recent advances in rna folding. *Journal of Biotechnology*, 261:97–104, Nov 2017.
6. H.-K. Cheong. Rapid preparation of rna samples for nmr spectroscopy and x-ray crystallography. *Nucleic Acids Research*, page e84–e84, Jun 2004.
7. Sebastian M Fica and Kiyoshi Nagai. Cryo-electron microscopy snapshots of the spliceosome: structural insights into a dynamic ribonucleoprotein machine. *Nature Structural & Molecular Biology*, 24(10):791–799, Oct 2017.
8. D. H. Mathews, W. N. Moss, and D. H. Turner. Folding and finding rna secondary structure. *Cold Spring Harbor Perspectives in Biology*, 2(12):a003665–a003665, Dec 2010.
9. Jinsong Zhang, Yuhan Fei, Lei Sun, and Qiangfeng Cliff Zhang. Advances and opportunities in rna structure experimental determination and computational modeling. *Nature Methods*, 19(10):1193–1207, Oct 2022.
10. Michael Zuker. On finding all suboptimal foldings of an rna molecule. *Science*, 244(4900):48–52, Apr 1989.
11. Jessica S Reuter and David H Mathews. Rnastructure: software for rna secondary structure prediction and analysis. *BMC Bioinformatics*, 11(1), Dec 2010.
12. Marc Parisien and François Major. The mc-fold and mc-sym pipeline infers rna structure from sequence data. *Nature*, 452(7183):51–55, Mar 2008.
13. Liang Huang, He Zhang, Dezhong Deng, Kai Zhao, Kaibo Liu, David A Hendrix, and David H Mathews. Linearfold: linear-time approximate rna folding by 5'-to-3' dynamic programming and beam search. *Bioinformatics*, page i295–i304, Jul 2019.
14. Jakob Hull Havgaard and Jan Gorodkin. Rna structural alignments, part i: Sankoff-based approaches for structural alignments. *RNA Sequence, Structure, and Function: Computational and Bioinformatic Methods*, pages 275–290, 2014.
15. Yinghan Fu, Gaurav Sharma, and David H. Mathews. Dynalign ii: common secondary structure prediction for rna homologs with domain insertions. *Nucleic Acids Research*, 42(22):13939–13948, Dec 2014.
16. Elena Rivas, Jody Clements, and Sean R Eddy. A statistical test for conserved rna structure shows lack of evidence for structure in lncrnas. *Nature Methods*, 14(1):45–48, Jan 2017.
17. Linyu Wang, Yuanning Liu, Xiaodan Zhong, Haiming Liu, Chao Lu, Cong Li, and Hao Zhang. Dmfold: a novel method to predict rna secondary structure with pseudoknots based on deep learning and improved base pair maximization principle. *Frontiers in genetics*, 10:143, 2019.
18. Xinshi Chen, Yu Li, Ramzan Umarov, Xin Gao, and Le Song. Rna secondary structure prediction by learning unrolled algorithms. *International Conference on Learning Representations,International Conference on Learning Representations*, Apr 2020.
19. Laiyi Fu, Yingxin Cao, Jie Wu, Qinke Peng, Qing Nie, and Xiaohui Xie. Ufold: fast and accurate rna secondary structure prediction with deep learning. *Nucleic acids research*, 50(3):e14–e14, 2022.
20. Kengo Sato, Manato Akiyama, and Yasubumi Sakakibara. Rna secondary structure prediction using deep learning with thermodynamic integration. *Nature communications*, 12(1):941, 2021.
21. Jaswinder Singh, Jack Hanson, Kuldip Paliwal, and Yaoqi Zhou. Rna secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. *Nature Communications*, Nov 2019.
22. Jaswinder Singh, Kuldip Paliwal, Tongchuan Zhang, Jaspreet Singh, Thomas Litfin, and Yaoqi Zhou. Improved rna secondary structure and tertiary base-pairing prediction using evolutionary profile, mutational coupling and two-dimensional transfer learning. *Bioinformatics*, 37(17):2589–2600, 2021.
23. Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19830–19843, October 2023.
24. Bowen Jing, Ezra Erives, Peter Pao-Huang, Gabriele Corso, Bonnie Berger, and Tommi Jaakkola. Eigenfold: Generative protein structure prediction with diffusion models. *arXiv preprint arXiv:2304.02198*, 2023.
25. Shuxin Zheng, Jiyan He, Chang Liu, Yu Shi, Ziheng Lu, Weitao Feng, Fusong Ju, Jiaxi Wang, Jianwei Zhu, Yaosen Min, et al. Towards predicting equilibrium distributions for molecular systems with deep learning. *arXiv preprint arXiv:2306.05445*, 2023.
26. Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12454–12465. Curran Associates, Inc., 2021.
27. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science,Lecture Notes in Computer Science*, Jan 2015.
28. Jiayang Chen, Zhihang Hu, Siqi Sun, Qingxiong Tan, Yixuan Wang, Qinze Yu, Licheng Zong, Liang Hong, Jin Xiao, Irwin King, et al. Interpretable rna foundation model from unannotated data for highly accurate rna structure and function predictions. *arXiv preprint arXiv:2204.00300*, 2022.
29. Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.
30. Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR, 18–24 Jul 2021.
31. Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.
32. Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.
33. Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10696–10706, 2022.

34. Yikun Zhang, Mei Lang, Jiuhong Jiang, Zhiqiang Gao, Fan Xu, Thomas Litfin, Ke Chen, Jaswinder Singh, Xiansong Huang, Guoli Song, et al. Multiple sequence alignment-based rna language model and its application to structural inference. *Nucleic Acids Research*, 52(1):e3–e3, 2024.
35. Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.
36. Hao Zhang, Chunhe Zhang, Zhi Li, Cong Li, Xu Wei, Borui Zhang, and Yuanning Liu. A new method of rna secondary structure prediction based on convolutional neural network and dynamic programming. *Frontiers in Genetics*, May 2019.
37. Zhen Tan, Yinghan Fu, Gaurav Sharma, and David H. Mathews. Turbofold ii: Rna structural alignment and secondary structure prediction informed by multiple homologs. *Nucleic Acids Research*, page 11570–11581, Nov 2017.
38. Michael F. Sloma and David H. Mathews. Exact calculation of loop formation probability identifies folding motifs in rna secondary structures. *RNA*, 22(12):1808–1818, Dec 2016.
39. Padideh Danaee, Mason Rouches, Michelle Wiley, Dezhong Deng, Liang Huang, and David Hendrix. bprna: large-scale automated annotation and analysis of rna secondary structure. *Nucleic acids research*, 46(11):5381–5394, 2018.
40. Ioanna Kalvari, Joanna Argasinska, Natalia Quinones-Olvera, Eric P Nawrocki, Elena Rivas, Sean R Eddy, Alex Bateman, Robert D Finn, and Anton I Petrov. Rfam 13.0: shifting to a genome-centric resource for non-coding rna families. *Nucleic acids research*, 46(D1):D335–D342, 2018.
41. Zasha Weinberg, Christina E Lünse, Keith A Corbino, Tyler D Ames, James W Nelson, Adam Roth, Kevin R Perkins, Madeline E Sherlock, and Ronald R Breaker. Detection of 224 candidate structured rnas by comparative analysis of specific subsets of intergenic regions. *Nucleic acids research*, 45(18):10811–10823, 2017.
42. Peter W Rose, Andreas Prlić, Ali Altunkaya, Chunxiao Bi, Anthony R Bradley, Cole H Christie, Luigi Di Costanzo, Jose M Duarte, Shuchismita Dutta, Zukang Feng, et al. The rcsb protein data bank: integrative view of protein, gene and 3d structural information. *Nucleic acids research*, page gkw1000, 2016.
43. Limin Fu, Beifang Niu, Zhengwei Zhu, Sitao Wu, and Weizhong Li. Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 28(23):3150–3152, 2012.
44. Chuong B. Do, Daniel A. Woods, and Serafim Batzoglou. Contrafold: Rna secondary structure prediction without physics-based models. *Bioinformatics*, page e90–e98, Jul 2006.
45. Kangkun Mao, Jun Wang, and Yi Xiao. Prediction of rna secondary structure with pseudoknots using coupled deep neural networks. *Biophysics Reports*, 6:146–154, 2020.
46. Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
47. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
48. M. Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, page 3406–3415, Jul 2003.
49. Ronny Lorenz, Stephan H Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. Viennarna package 2.0. *Algorithms for molecular biology*, 6:1–14, 2011.
50. Shay Zakov, Yoav Goldberg, Michael Elhadad, and Michal Ziv-Ukelson. Rich parameterization improves rna structure prediction. *Journal of Computational Biology*, 18(11):1525–1542, 2011.
51. Eric P Nawrocki, Sarah W Burge, Alex Bateman, Jennifer Daub, Ruth Y Eberhardt, Sean R Eddy, Evan W Floden, Paul P Gardner, Thomas A Jones, John Tate, et al. Rfam 12.0: updates to the rna families database. *Nucleic acids research*, 43(D1):D130–D137, 2015.
52. Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402, 1997.
53. Eric P. Nawrocki and Sean R. Eddy. Infernal 1.1: 100-fold faster rna homology searches. *Bioinformatics*, page 2933–2935, Nov 2013.
54. Kévin Darty, Alain Denise, and Yann Ponty. Varna: Interactive drawing and editing of the rna secondary structure. *Bioinformatics*, 25(15):1974, 2009.
55. Martín González Buitrón, Ronaldo Romario Tunque Cahui, Emilio García Ríos, Layla Hirsh, Gustavo Parisi, María Silvina Fornasari, and Nicolas Palopoli. Codnas-rna: a database of conformational diversity in the native state of rna. *Bioinformatics*, 38(6):1745–1748, 2022.