OXFORD

Briefings in Bioinformatics, 2025, **26(1)**, bbae618
https://doi.org/10.1093/bib/bbae618
Problem Solving Protocol

RNADiffFold: generative RNA secondary structure prediction using discrete diffusion models

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

[1]Hangzhou Institute of Medicine, Chinese Academy of Sciences, Hangzhou 310018, Zhejiang, China
[2]College of Information Engineering, Zhejiang University of Technology, Hangzhou 310014, Zhejiang, China
[3]College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310014, Zhejiang, China
[4]Hangzhou Institute for Advanced Study, University of Chinese Academy of Sciences, Hangzhou 310024, Zhejiang, China

*Corresponding authors. Pengju Yan, Hangzhou Institute of Medicine, Chinese Academy of Sciences, Hangzhou, 310018, Zhejiang, China.
E-mail: yanpengju@gmail.com; Xiaolin Li, Hangzhou Institute of Medicine, Chinese Academy of Sciences, Hangzhou, 310018, Zhejiang, China.
E-mail: xiaolinli@ieee.org

‡Zhen Wang and Yizhen Feng contributed equally to this work and should be considered co-first authors.

Abstract

Ribonucleic acid (RNA) molecules are essential macromolecules that perform diverse biological functions in living beings. Precise prediction of RNA secondary structures is instrumental in deciphering their complex three-dimensional architecture and functionality. Traditional methodologies for RNA structure prediction, including energy-based and learning-based approaches, often depict RNA secondary structures from a static perspective and rely on stringent a priori constraints. Inspired by the success of diffusion models, in this work, we introduce RNADiffFold, an innovative generative prediction approach of RNA secondary structures based on multinomial diffusion. We reconceptualize the prediction of contact maps as akin to pixel-wise segmentation and accordingly train a denoising model to refine the contact maps starting from a noise-infused state progressively. We also devise a potent conditioning mechanism that harnesses features extracted from RNA sequences to steer the model toward generating an accurate secondary structure. These features encompass one-hot encoded sequences, probabilistic maps generated from a pre-trained scoring network, and embeddings and attention maps derived from RNA foundation model. Experimental results on both within- and cross-family datasets demonstrate RNADiffFold's competitive performance compared with current state-of-the-art methods. Additionally, RNADiffFold has shown a notable proficiency in capturing the dynamic aspects of RNA structures, a claim corroborated by its performance on datasets comprising multiple conformations.

Keywords: RNA secondary structure prediction; deep learning; discrete diffusion model

Introduction

Ribonucleic acid (RNA) is a vital biomolecule with diverse roles beyond the transfer of genetic information from DNA to proteins. It is involved in catalysis, regulation, and protein synthesis, primarily through its non-coding regions [1]. For instance, microRNAs (miRNAs) regulate gene expression post-transcriptionally, with their dysregulation linked to various diseases [2]. Long non-coding RNAs (lncRNAs) are crucial in cellular processes like chromatin modification and transcriptional regulation [3]. Small nuclear RNAs (snRNAs) form spliceosome complexes, influencing splicing patterns and mRNA abundance [4]. Thus, delving into RNA functionality is essential for unraveling 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 secondary structure of RNA 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 its single-stranded nature, RNA's 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, understanding RNA's secondary structure is key to deciphering its biological mechanisms and progressing to tertiary structure resolution [8].

Over the past decades, researchers have developed various methods for RNA secondary structure prediction, combining experimental and computational approaches. These methods fall into three main categories: energy-based, covariation-based, and deep learning-based [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 [10–13].

Received: June 7, 2024. **Revised:** October 12, 2024. **Accepted:** November 18, 2024

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 infer secondary structures by considering the co-evolutionary relationships between RNA sequences and structures [14–16]. 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 accumulation of RNA secondary structure data and the rapid development of deep learning technologies, deep learning-based RNA secondary structure prediction methods have gained traction. These methods employ neural networks, such as Bi-LSTM, Transformer, and U-Net to calculate base pairing probabilities to capture long-range interactions [17–19]. Some approaches integrate thermodynamic knowledge [20], adopt transfer learning strategies [21], or incorporate evolutionary and mutational coupling information [22] to optimize prediction results and alleviate prediction biases. However, existing deep learning-based methods still exhibit shortcomings in model generalization performance, especially when modeling unknown RNA families. This limitation arises because their model parameters often derive from a limited pool of known structures, thereby restricting their adaptability to new data. To overcome limitations in training data quantity and distribution and enhance model generalization capabilities, new computational methods are necessary to achieve more accurate and comprehensive outcomes in RNA secondary structure prediction.

In recent years, diffusion models have demonstrated outstanding performance in various prediction tasks [23–25]. Inspired by this, we introduce **RNADiffFold**, a novel framework for RNA secondary structure prediction using discrete diffusion models. The framework aims to predict a deterministic RNA secondary structure in a generative manner. RNADiffFold first represents RNA secondary structures as binary contact maps. 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. This approach simplifies the complex RNA secondary structure prediction task into a pixel-level image segmentation task.

RNADiffFold comprises two main components: the diffusion model and conditional control. The diffusion model component is based on discrete data space multinomial diffusion [26]. As illustrated in Fig. 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, we employ U-Net [27] as the learning network and add 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 high-dimensional embeddings and attention maps from RNA foundation model (RNA-FM) [28], with dimensionality reduction through different MLPs. At each time step of the inverse diffusion process, all these sequence features are fused with the intermediate state x_t. This design enables RNADiffFold to leverage the powerful capabilities of the diffusion model to predict RNA secondary structures while integrating various sequence features to enhance the accuracy and stability of predictions.

Building upon previous work [19, 21, 22], we evaluate 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, RNADiffFold not only predicts static RNA secondary structures but also captures dynamic multi-conformational features to some extent by learning the distribution of secondary structure conformations.

Materials and methods

Preliminaries

Diffusion models

Diffusion models [29–30] 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 the 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 [26, 31, 32] are proposed to generate high-dimension discrete data. For example, D3PM [31] considers a transition matrix with an absorbing state or using discretized, truncated Gaussian distribution. VQ-diffusion [32] accommodates discrete data with a lazy random walk or a random masking operation.

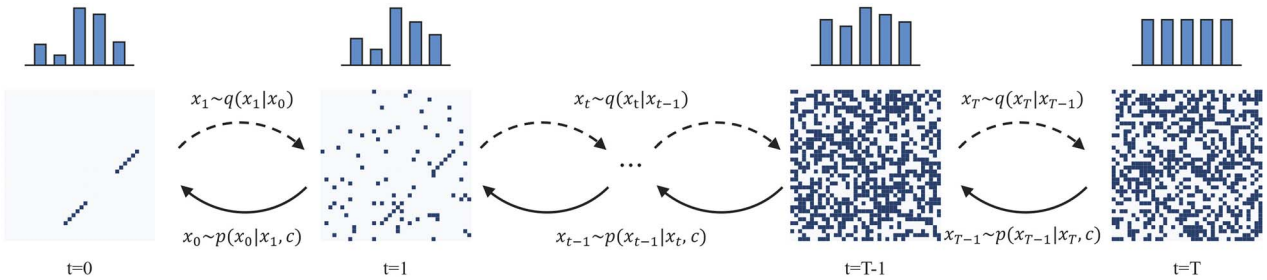

Figure 1. **Overview of RNA secondary structure prediction with multinomial diffusion.** During the diffusion process $q(x_t|x_{t-1})$, discrete noise is gradually introduced to corrupt the contact map from left to right. In the denoise process $p_\theta(x_{t-1}|x_t, c)$, a generative model learns to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

RNA foundation model

RNA-FM [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: pre-training and fine-tuning. In the pre-training 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 pre-training 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 pre-trained 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.

It is worth noting that a recent study [33] 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

As depicted in Fig. 2A, RNADiffFold consists of the diffusion model component and the condition construction unit. 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.

Diffusion process

As illustrated in Fig. 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 [34], discrete noise is gradually added to the sample at each timestep t through the forward process defined as follows:

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

where x_t represents a contact matrix at time t in one-hot encoded format $x_t^{ij} \in \{0,1\}^K (K = 2, 0 \le i,j \le L)$. $\vec{1}$ is an all-one matrix. β_t is the chance of resampling another pairing possibility uniformly. We apply the cosine schedule [30] 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:

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

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

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

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

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

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.

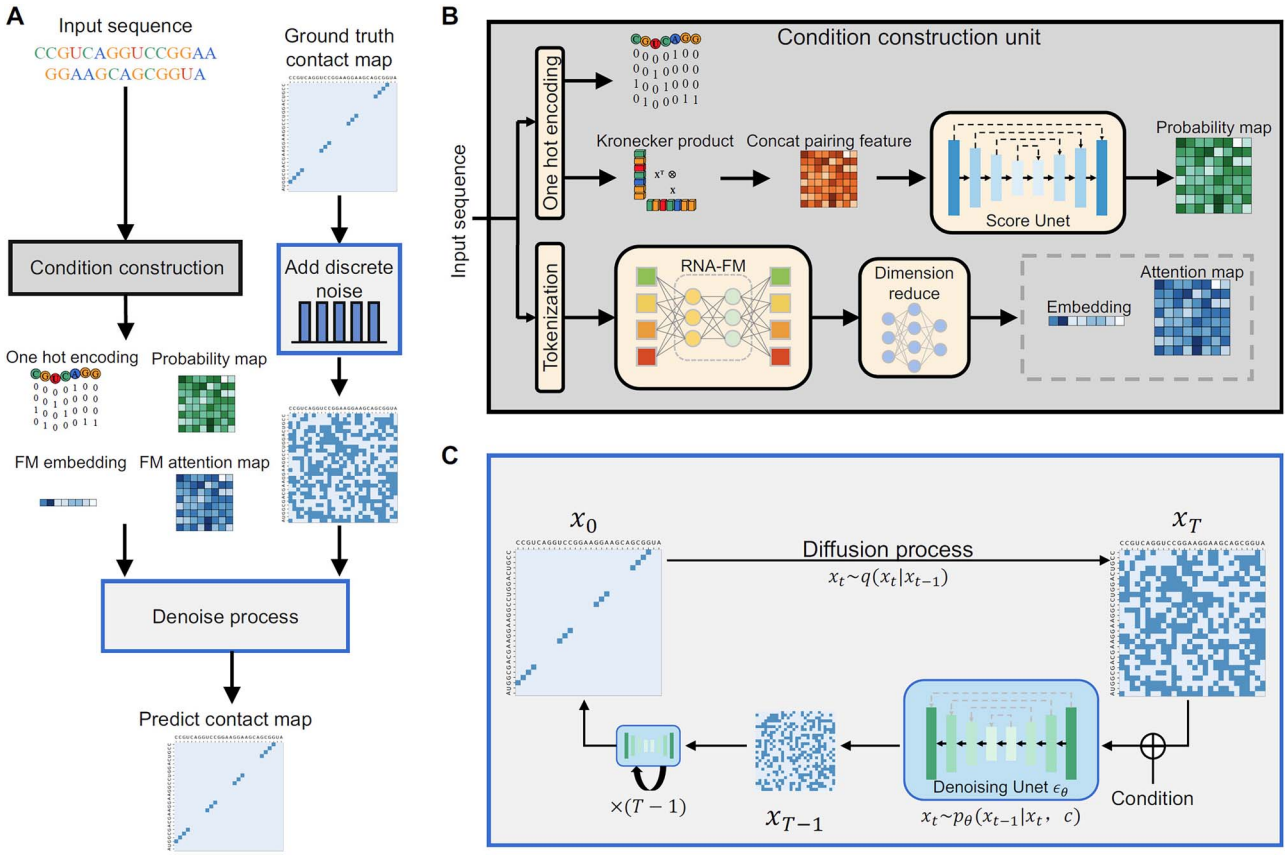

Figure 2. **The pipeline of RNADiffFold.** (**A**) Overview of RNADiffFold workflow as a supervised task. Given the input sequence, we construct the representation through the condition construction stage. During the training stage, discrete noise is incrementally added to the ground truth contact map. 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 U-Net network as the denoising network for training and inferring.

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

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

Furthermore, $\mathcal{L}_T$ corresponds to the second term in equation 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 U-Net network are illustrated in Supplementary Fig. S1. Benefiting from the flexibility of U-Net to handle inputs of different sizes, we can also process variable-length sequence data.

Condition construction unit

The condition construction is depicted in Fig. 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 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 Fig. 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 [35], 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 [33], 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

We evaluated RNADiffFold using several datasets for both within-family and cross-family scenarios, following prior work [19, 21, 22]. These datasets include RNAStrAlign [36], ArchiveII [37], bpRNA-1m [38], bpRNA-new [20], and PDB. 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 3975 sequences spanning 10 RNA families. Due to their similar family distributions, RNAStrAlign is often used as a training set and ArchiveII as a test set. The bpRNA-1m dataset contains 102 318 sequences from 2588 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 [39] and includes 1500 new RNA families detected by newly developed techniques [40]. Furthermore, the PDB dataset, collected from the Protein Data Bank [41], comprises high-resolution RNA structures with resolutions less than 3.5Å (as of 9 April 2020).

Before using the datasets, we performed preprocessing steps to enhance computational efficiency and reduce redundancy. Specifically, for the RNAStrAlign dataset, we removed sequences longer than 640 nucleotides and randomly split the dataset into training, validation, and test sets in an 8:1:1 ratio. Redundant sequences were removed using methods from E2Efold [18] and Ufold [19]. 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, we used CD-HIT-EST [42] to remove sequences with over 80% similarity, following the methods of MXfold2 [20] and SPOT-RNA [21]. The remaining data, referred to as bpRNA, were divided into training, validation, and test sets labeled TR0, VL0, and TS0, respectively. For the bpRNA-new dataset, sequences with over 80% similarity were removed using CD-HIT-EST, and sequences longer than 500 nucleotides were filtered out. For the PDB dataset, we applied the same dataset division as SPOT-RNA2 [22], resulting in training (TR1), validation (VL1), and various test sets (TS1, TS2, TS3, TS-hard).

Furthermore, we analyzed the relationship between sequence length and the number of base pairs in each dataset, finding a linear relationship for most samples, as shown in Supplementary Fig. S2. 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. Thus, sequences longer than 200 nucleotides with fewer than 10 base pairs were removed from the training set to reduce the impact of low-pairing sequences. Validation and test sets remained consistent with previous work. 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 data augmentation similar to Ufold [19]. In the bpRNA-new dataset, 20 – 30% of nucleotides were randomly mutated to create new data. Then, sequences with over 80% similarity to real sequences were removed using CD-HIT-EST, yielding 2717 mutated sequences (named Mutate-seq) for supplementary training. Labels for Mutate-seq were generated using Contrafold [43], a probabilistic method extending stochastic context-free grammars (SCFGs) with discriminative training objectives and flexible feature representations.

In summary, the dataset configuration for this study includes multiple training sets derived from RNAStrAlign, bpRNA TR0, and Mutate-seq, along with TR1 from the PDB dataset utilized for cross-family fine-tuning. For within-family evaluations, the test sets include sequences from ArchiveII and bpRNA TS0. Conversely, cross-family testing employs bpRNA-new and multiple test sets from PDB (TS1, TS2, TS3, and TS-hard). The detailed statistics of the final datasets employed for the training, validation, and testing phases are systematically presented in Supplementary Table S1.

We also 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, 3202 sequences between 160 and 320 nucleotides, and 3119 sequences between 320 and 640 nucleotides), as shown in Supplementary Fig. S3A, which hindered the model's ability to learn long-range RNA interactions. To address this issue, a data balancing strategy [44] 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 48 658. This initiative effectively mitigated the impact of data imbalance on model training. Supplementary Figure S3B shows the distribution of sequence length and number of base pairs after data balance. In addition, to improve the training efficiency, we also developed a data bucketing strategy to process the above datasets, and the details are given in Supplementary Section 1.1.

Experiments and results

Evaluation criteria and experimental setup

To comprehensively evaluate the prediction accuracy of RNADiff-Fold, 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 [45]. We set the number of diffusion steps T as 20 during the diffusion process, considering that the contact map contains sparse classification information. A large T, on the contrary, tends to degrade prediction performance. Adam [46] is employed as the optimizer with a learning rate of 1e-3. The model undergoes training for a maximum of 400 epochs, with validation performed every 20 epochs. Early stopping with a patience of 5 is implemented to prevent overfitting. Supplementary Table S10 details all the hyperparameters of the experiment.

The training process is divided into two stages. Firstly, the model is 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 is performed on the PDB dataset, and testing is 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 an Intel(R) Xeon(R) Gold 6326 CPU @ 2.90GHz. An analysis of computational efficiency is provided in Supplementary Section 1.3 and Table S12.

Baselines

We compared RNADiffFold with 10 competitive baseline methods for RNA secondary structure prediction, including 6 energy-based approaches and 4 deep learning-based methods. Among the energy-based methods, Mfold [47] utilizes nearest-neighbor energy parameters and dynamic programming to find the structure with the lowest energy. Linearfold [13] combines dynamic programming with beam search to enhance efficiency. RNAfold [48] integrates dynamic programming algorithms with a thermodynamic-based energy model to predict optimal RNA structures. RNAStructure [11] calculates minimum free energy and optimizes prediction results based on experimental data. Contrafold [43] uses conditional log-linear models, extending to SCFGs, and integrates thermodynamic parameters. ContextFold [49] introduces a fine-grained model with a large parameter set (approximately 70 000).

For the deep learning-based methods, SPOT-RNA [21] employs a CNN and BiLSTM architecture and utilizes transfer learning to enhance performance. MXFold2 [20] combines deep learning with thermodynamic parameters for more accurate predictions. E2Efold [18] innovatively integrates the Transformer model with an unrolling algorithm, imposing hard constraints on RNA secondary structure. UFold [19] utilizes U-Net and a structure similar to image inputs to capture long-range dependencies effectively. Due to the lack of reproducible results from SPOT-RNA and ContextFold in experiments, we referenced the experimental data of UFold for comparison analysis.

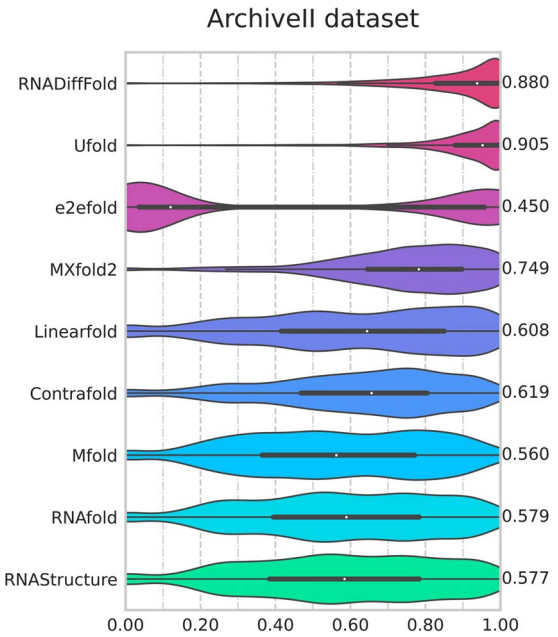

Figure 3. **Violin plot on the ArchiveII dataset.** Visualization of F1 value of RNADiffFold against other methods.

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, evaluation results on these datasets effectively reflect the model's predictive performance. As shown in Fig. 3 and Supplementary Table S2, for the ArchiveII test set, RNADiffFold demonstrates good predictive performance with an F1 score of 0.880, surpassing all energy-based methods and ranking among the top in deep learning-based methods. Its F1 score is slightly lower than UFold and RNA-FM, possibly due to information loss from 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 the condition, as further confirmed in subsequent ablation experiments.

In contrast, the bpRNA dataset from Rfam12.2 [50] encompasses a broader range of RNA families. Evaluation results on its test set, TS0, show RNADiffFold achieves an average F1 score of 0.711, surpassing all energy-based and deep learning-based methods (Fig. 4 and Supplementary Table S2). 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 highlights the superiority of RNADiffFold in the field of RNA secondary structure prediction, demonstrating that its outstanding performance results from its unique algorithm design and feature fusion strategy, rather than a simple combination of the UFold scoring network and RNA-FM output features.

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. 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

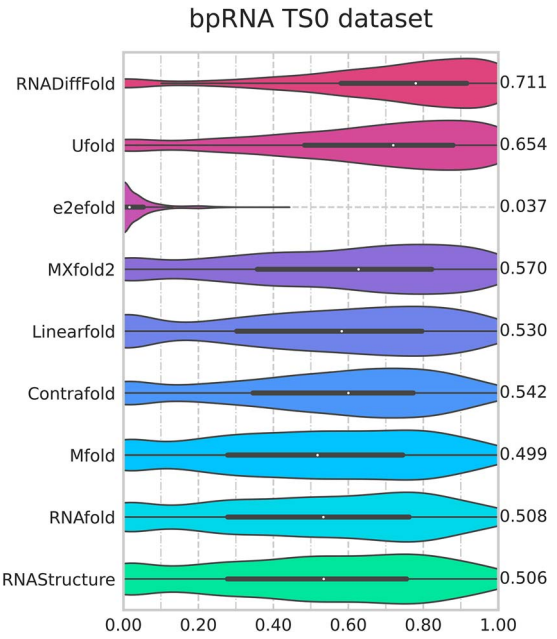

Figure 4. **Violin plot on the bpRNA TS0 dataset.** Visualization of F1 value of RNADiffFold against other methods.

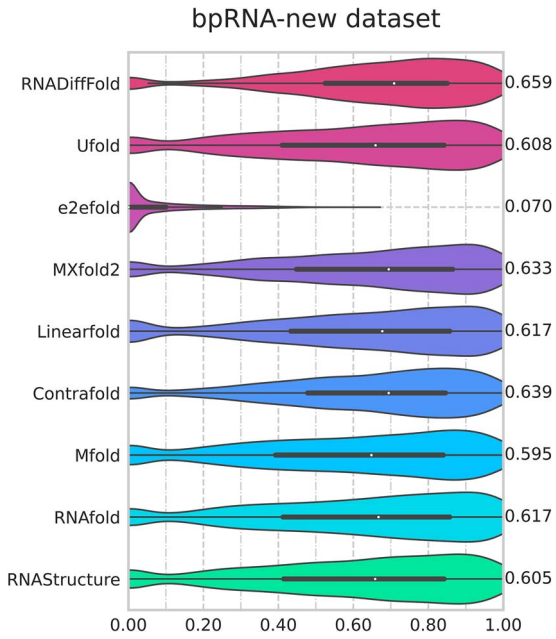

Figure 5. **Violin plot on the bpRNA-new dataset.** Visualization of F1 value of RNADiffFold against other methods.

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]

samples from a total of 1304 that met the criteria for precision, recall, and F1 score calculations. Comparing RNADiffFold with other mainstream methods, the results, presented in Table 1 and Supplementary Fig. S4, demonstrate RNADiffFold's outstanding performance on long-range base pair RNA data. Compared to UFold, RNADiffFold's predicted precision and recall are closer, indicating a more stable predictive performance. This stability likely stems from the effective integration of the pre-trained UFold scoring network with RNA-FM output features, providing RNADiffFold with rich contextual and structural information, thereby enhancing its accuracy in predicting long-range base pairs.

Performance on cross-family datasets

The "cross-family" datasets indicate that the RNA family species in the test set are entirely distinct from those in the training set. Performance on such datasets, particularly for deep learning-based methods, better reflects their understanding of RNA folding patterns and their ability to generalize to unknown families. To enhance the model's generalization performance, we adopted UFold's data augmentation strategy, generating 2717 mutated sequences with mutation rates between 20 and 30% from the original sequences. These sequences were predicted for their secondary structures by Contrafold, serving as pseudo-labels. These mutated datasets were then integrated into RNAStrAlign and TR0 for training and comprehensively evaluated on three different test sets.

As shown in Fig. 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 incorporates thermodynamic constraints, or UFold, which employs biological-based hard constraints, RNADiffFold predicts the secondary structures of new, unknown family sequences by learning the distribution of RNA conformations. Additionally, we analyzed the performance across different sequence lengths by partitioning the bpRNA-new dataset at 100 nt intervals. As shown in Supplementary Fig. 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, possibly due to the absence of constraints.

To comprehensively evaluate RNADiffFold in cross-family scenarios, we conducted additional validation using the PDB dataset. The PDB dataset [41] contains secondary structure information extracted from high-resolution RNA 3D structures, providing a reliable basis for evaluation. Following the same dataset partitioning method as SPOT-RNA2 [22], we fine-tuned the pre-trained model on the PDB dataset. Figure 6 shows that RNADiffFold achieved an average F1 score of 0.736, demonstrating comparable performance with other methods. Further analysis on different test sets within PDB (TS1, TS2, and TS3) is detailed in Supplementary Table S4 and Fig. S6. We also constructed a more challenging test set, TS-hard, which was curated from TS1 and TS3 by excluding similar sequences using BLAST-N [51] and INFERNAL [52] models. The results on TS-hard, shown in Supplementary Table S5 and Fig. S7, validate that RNADiffFold maintains comparable performance with most methods even in challenging prediction tasks.

To further substantiate RNADiffFold's predictive capability, we employed two statistical significance assessment methods: an individual t-test-based method and a bootstrapping-based method. The results in Table S6 indicate that RNADiffFold significantly outperforms other methods statistically, with most

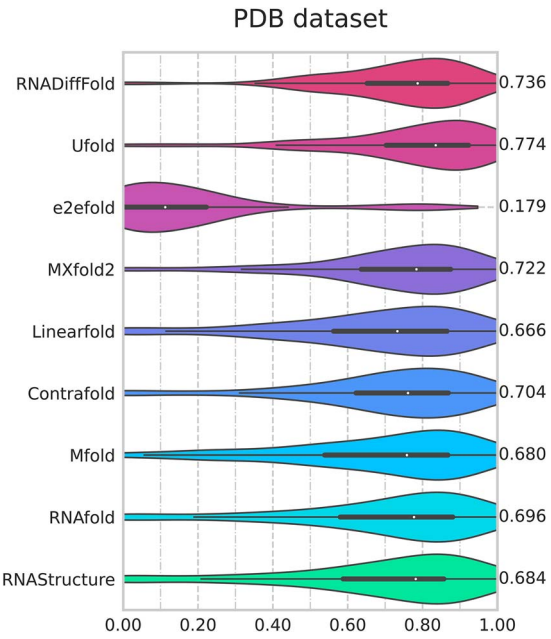

Figure 6. **Violin plot on the PDB test dataset encompassing TS1, TS2, and TS3.** Visualization of F1 value of RNADiffFold against other methods.

P-values being less than 0.05. For the relatively small-scale PDB dataset, bootstrapping proved to be a more effective evaluation method. Supplementary Table S7 and Fig. 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 our conclusions.

Multiple sampling and voting strategy testing experiments

To further enhance the model's performance, we implemented a strategy based on multiple sampling and voting. Given that RNADiffFold is a generative prediction method capable of learning the distribution of secondary structures, we used random seeds to generate a set of predicted structures. We then conducted cluster analysis by calculating the similarity between these predictions and selected the most frequently occurring structure from the largest cluster as the final prediction. When no clear clusters were present, a random selection method determined the final prediction. This strategy resulted in a performance improvement, as shown in Table S8. 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 multiple sampling and voting strategy helps reduce the uncertainty of model predictions, thereby enhancing the accuracy of RNA secondary structure prediction. The performance shown in Figs 3–6 all adopted this strategy.

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. The predicted results were converted into.ct format based on base pairing relationships and visualized using VARNA [53]. 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 [54], 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 extremely similar sequences, 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. Supplementary Figure S10 shows examples of 4 successfully predicted clusters, of which Cluster 189 represents the prediction of 9 similar clusters. The failed cases occurred because the ground truth sequences tended to be completely unpaired or contained pairings that violated standard pairing rules, making accurate predictions challenging, as shown in Supplementary Fig. S11.

Ablation study

In this section, we explore the impact of different conditioning strategies on the performance of RNADiffFold while maintaining 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}, 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 and Supplementary Table S9 present the comprehensive performance evaluation results of RNADiffFold on four test sets. Key observations drawn from this study are as follows: (i) 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 (version v1). However, the performance of the v1 is relatively limited when faced with sequences from unknown families, indicating the inadequacy of one-hot encoding in feature extraction. (ii) The probability map c_u from the Ufold scoring network (version v2) contributes more significantly to cross-family data, while c_{emb} and c_{attn} from RNA-FM (version v3) play a more significant role in within-family data. Integrating these output features as sequence condition information achieves more balanced performance on both within-family and cross-family data. (iii) Comparing versions v4 and v5 highlights the utility of the attention map c_{attn}. Introducing c_{attn} results in slight performance improvements on almost all datasets, albeit modest. This may be due to partial feature information loss from excessive dimensionality reduction. To determine if the exceptional performance of RNADiffFold primarily stems from its diffusion process, we conducted further comparative experiments detailed in Supplementary Section 1.2 and Table S11. The results confirm that the integration of features from Ufold and RNA-FM alone does not account for the superior outcomes observed; the diffusion process is pivotal to its success.

Additionally, early experiments explored the relationship between different diffusion step sizes and model performance. Unexpectedly, smaller step sizes performed better in the diffusion process, as shown in Supplementary Fig. S9. This may be because the secondary structure contact map contains sparser information compared to real-world images. 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.

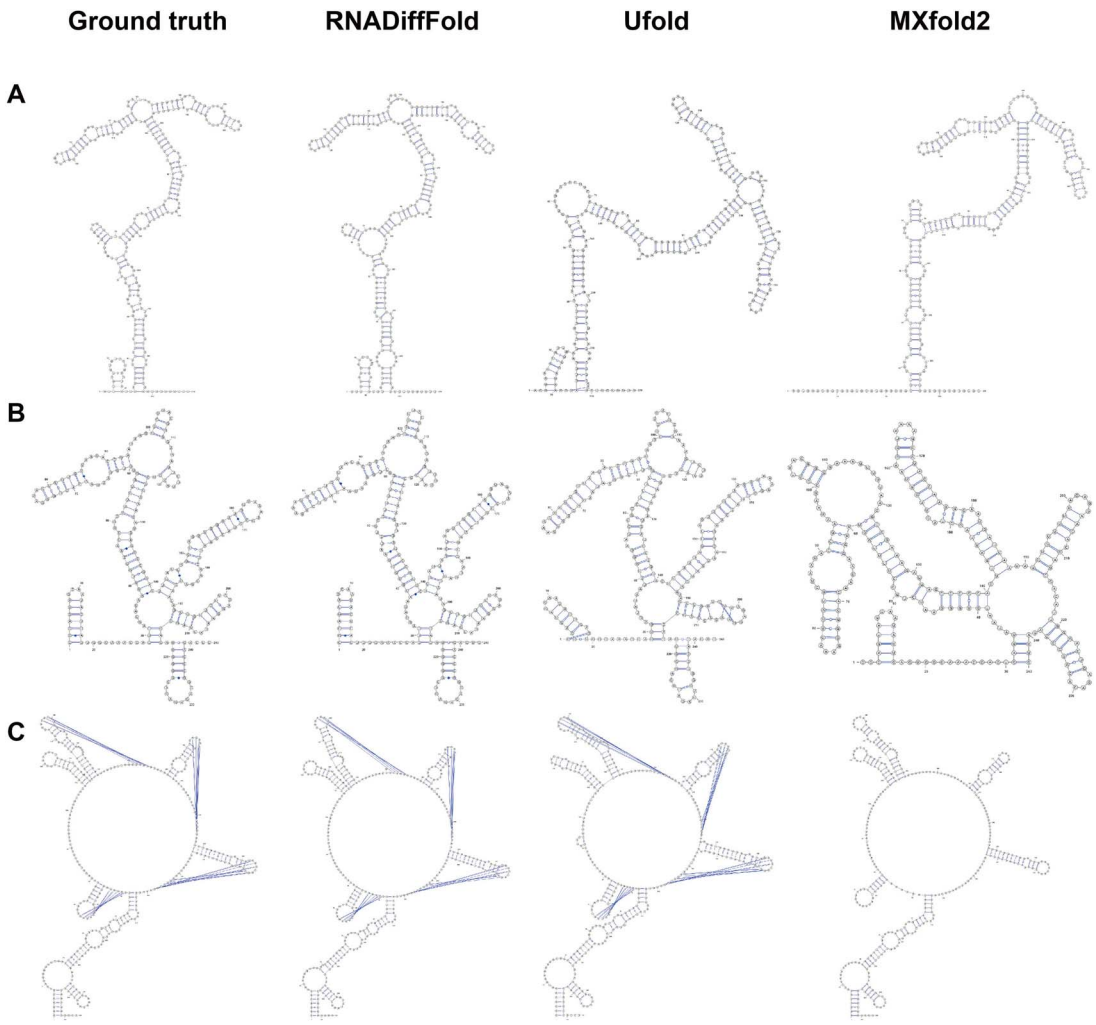

Figure 7. **Visualization compares three examples predicted by RNADiffFold with those from two other methods against the ground truth.** The RNA sequences are from the following families: (**A**) Aspergillus fumigatus, recorded in the SRPDB database; (**B**) Alphaproteobacteria subfamily 16S rRNA, recorded 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 detail than the other two methods.

Discussion

In this paper, we present RNADiffFold, a novel RNA secondary structure prediction method based on the discrete diffusion model. Unlike traditional deep learning-based prediction methods, RNADiffFold treats secondary structure prediction as a pixel-level contact map segmentation task, utilizing discrete diffusion processes to capture conformational distributions. During 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 process, RNADiffFold employs a U-Net as the denoising network. Additionally, we propose an effective conditioning strategy to extract condition information from RNA sequences to guide structure prediction.

RNADiffFold offers several significant advantages over previous methods. First, traditional approaches typically predict secondary structures in a deterministic manner, which contradicts the dynamic folding nature of RNA. RNADiffFold, however, explores the conformational space of RNA through diffusion processes without imposing any explicit hard constraints, allowing it to predict non-canonical pairings arising from tertiary interactions. Second, our proposed conditioning strategy leverages the strengths of different models. From another perspective, RNADiffFold can be viewed as a downstream task of pre-trained models, avoiding the need for embedding 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. Additionally, 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. First, 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 less category information, with most regions classified as “0” class. Therefore, resources should be reduced in these easily learnable regions. Second, adopting more efficient conditional pre-training models could enhance performance. Due to computational cost limitations, this study has not retrained a condition control model tailored for the diffusion model. Theoretically, using more suitable and thoroughly trained pre-training models to extract sequence

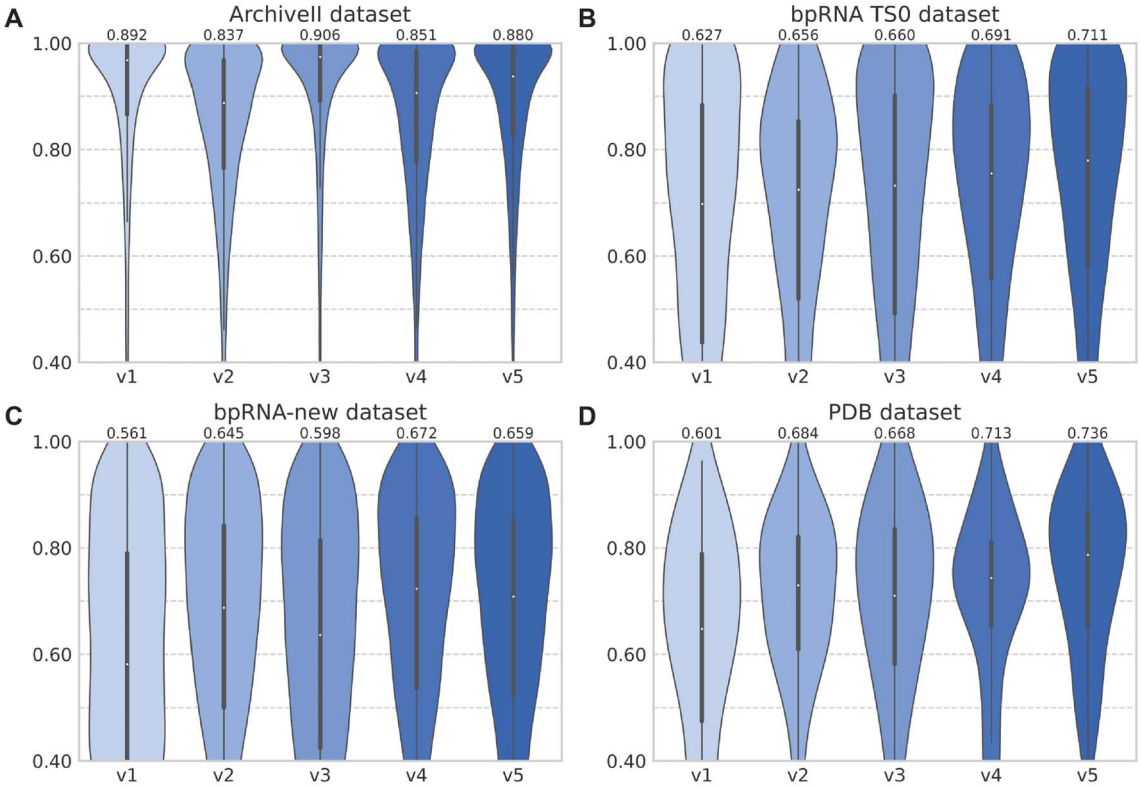

Figure 8. **F1 values on the four datasets** (within-family datasets ArchiveII (**A**), bpRNA TS0 (**B**) and cross-family datasets bpRNA-new (**C**), PDB datasets (**D**)). v1-v5 are different condition construction strategies combined with the diffusion process. v1: one-hot encoding c_{onehot}; v2: 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.

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.

Acknowledgements

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

Supplementary data

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

Author ContributionS

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

Conflict of interest

None declared.

Funding

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

Data Availability

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

References

1. Seetin MG, Mathews DH. RNA structure prediction: an overview of methods. In: Keiler K. (eds) *Bacterial Regulatory RNA*. Methods in Molecular Biology, Humana Press, Totowa, NJ, 2012;**905**:99–122. https://doi.org/10.1007/978-1-61779-949-5_8.

2. Hammond SM. An overview of microRNAs. *Adv Drug Deliv Rev* 2015;**87**:3–14. https://doi.org/10.1016/j.addr.2015.05.001.
3. Mercer TR, Dinger ME, Mattick JS. Long non-coding RNAs: insights into functions. *Nat Rev Genet* 2009;**10**:155–9. https://doi.org/10.1038/nrg2521.
4. Bratkovič T, Božič J, Rogelj B. Functional diversity of small nucleolar RNAs. *Nucleic Acids Res* 2020;**48**:1627–51. https://doi.org/10.1093/nar/gkz1140.
5. Fallmann J, Will S, Engelhardt J. *et al*. Recent advances in RNA folding. *J Biotechnol* 2017;**261**:97–104. https://doi.org/10.1016/j.jbiotec.2017.07.007.
6. Cheong H-K, Hwang E, Lee C. *et al*. Rapid preparation of RNA samples for NMR spectroscopy and x-ray crystallography. *Nucleic Acids Res* 2004;**32**:e84. https://doi.org/10.1093/nar/gnh081.
7. Fica SM, Nagai K. Cryo-electron microscopy snapshots of the spliceosome: structural insights into a dynamic ribonucleoprotein machine. *Nat Struct Mol Biol* 2017;**24**:791–9. https://doi.org/10.1038/nsmb.3463.
8. Mathews DH, Moss WN, Turner DH. Folding and finding RNA secondary structure. *Cold Spring Harb Perspect Biol* 2010;**2**:a003665–5. https://doi.org/10.1101/cshperspect.a003665.
9. Zhang J, Fei Y, Sun L. *et al*. Advances and opportunities in RNA structure experimental determination and computational modeling. *Nat Methods* 2022;**19**:1193–207. https://doi.org/10.1038/s41592-022-01623-y.
10. Zuker M. On finding all suboptimal foldings of an RNA molecule. *Science* 1989;**244**:48–52. https://doi.org/10.1126/science.2468181.
11. Reuter JS, Mathews DH. RNAstructure: software for RNA secondary structure prediction and analysis. *BMC Bioinformatics* 2010;**11**:1–9. https://doi.org/10.1186/1471-2105-11-129.
12. Parisien M, Major F. The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature* 2008;**452**:51–5. https://doi.org/10.1038/nature06684.
13. Huang L, Zhang H, Deng D. *et al*. Linearfold: linear-time approximate RNA folding by 5'-to-3' dynamic programming and beam search. *Bioinformatics* 2019;**35**:i295–304. https://doi.org/10.1093/bioinformatics/btz375.
14. Havgaard JH, Gorodkin J. RNA structural alignments, part I: Sankoff-based approaches for structural alignments. In: Gorodkin J, Ruzzo W. (eds) *RNA Sequence, Structure, and Function: Computational and Bioinformatic Methods*. Methods in Molecular Biology, Humana Press, Totowa, NJ, 2014;**1097**:275–290. https://doi.org/10.1007/978-1-62703-709-9_13.
15. Yinghan F, Sharma G, Mathews DH. Dynalign II: common secondary structure prediction for RNA homologs with domain insertions. *Nucleic Acids Res* 2014;**42**:13939–48. https://doi.org/10.1093/nar/gku1172.
16. Rivas E, Clements J, Eddy SR. A statistical test for conserved RNA structure shows lack of evidence for structure in lncRNAs. *Nat Methods* 2017;**14**:45–8. https://doi.org/10.1038/nmeth.4066.
17. Wang L, Liu Y, Zhong X. *et al*. Dmfold: a novel method to predict RNA secondary structure with pseudoknots based on deep learning and improved base pair maximization principle. *Front Genet* 2019;**10**:143. https://doi.org/10.3389/fgene.2019.00143.
18. Chen X, Yu L, Umarov R. *et al*. RNA secondary structure prediction by learning unrolled algorithms. In: *International Conference on Learning Representations*, Toulon, France: ICLR publisher, 2020.
19. Laiyi F, Cao Y, Jie W. *et al*. Ufold: fast and accurate RNA secondary structure prediction with deep learning. *Nucleic Acids Res* 2022;**50**:e14–4.
20. Sato K, Akiyama M, Sakakibara Y. RNA secondary structure prediction using deep learning with thermodynamic integration. *Nat Commun* 2021;**12**:941. https://doi.org/10.1038/s41467-021-21194-4.
21. Singh J, Hanson J, Paliwal K. *et al*. RNA secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. *Nat Commun* 2019;**10**:5407. https://doi.org/10.1038/s41467-019-13395-9.
22. Singh J, Paliwal K, Zhang T. *et al*. Improved RNA secondary structure and tertiary base-pairing prediction using evolutionary profile, mutational coupling and two-dimensional transfer learning. *Bioinformatics* 2021;**37**:2589–600. https://doi.org/10.1093/bioinformatics/btab165.
23. Chen S, Sun P, Song Y. *et al*. DiffusionDet: diffusion model for object detection. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Paris, France: ICCV, pp. 19830–43, 2023.
24. Jing B, Erives E, Pao-Huang P. *et al*. Eigenfold: generative protein structure prediction with diffusion models. In: *International Conference on Learning Representations*, MLDD workshop, Kigali, Rwanda: ICLR publisher, 2023. https://openreview.net/forum?id=BgbRVzfQqFp.
25. Zheng S, He J, Liu C. *et al*. Towards predicting equilibrium distributions for molecular systems with deep learning. Nat Mach Intell 2024;**6**:558-567.
26. Hoogeboom E, Nielsen D, Jaini P. *et al*. Argmax flows and multinomial diffusion: learning categorical distributions. In: Ranzato M, Beygelzimer A, Dauphin Y. *et al*. (eds.), *Advances in Neural Information Processing Systems*, Red Hook, New York, USA, Vol. **34**. Curran Associates, Inc., 2021, 12454–65.
27. Ronneberger O, Fischer P, Brox. U-net: convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells W, Frangi A. (eds) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Munich, Germany: Springer International Publishing, 2015;**9351**. https://doi.org/10.1007/978-3-319-24574-4_28.
28. Chen J, Hu Z, Sun S. *et al*. Interpretable RNA foundation model from unannotated data for highly accurate RNA structure and function predictions. In: *International Conference on Machine Learning*, Baltimore, Maryland, USA, workshop. PMLR, 2022.
29. Sohl-Dickstein J, Weiss E, Maheswaranathan N. *et al*. Deep unsupervised learning using nonequilibrium thermodynamics. In: *International Conference on Machine Learning*, Lille France, pp. 2256–65. PMLR, 2015.
30. Nichol AQ, Dhariwal P. Improved denoising diffusion probabilistic models. In: *International Conference on Machine Learning*, pp. 8162–71. PMLR, 2021.
31. Austin J, Johnson DD, Ho J. *et al*. Structured denoising diffusion models in discrete state-spaces. *Adv Neural Inf Process Syst* 2021;**34**:17981–93.
32. Shuyang G, Dong C, Bao J. *et al*. Vector quantized diffusion model for text-to-image synthesis. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, Louisiana, USA: IEEE, 10696–706, 2022.
33. Zhang Y, Lang M, Jiang J. *et al*. Multiple sequence alignment-based RNA language model and its application to structural inference. *Nucleic Acids Res* 2024;**52**:e3–3. https://doi.org/10.1093/nar/gkad1031.
34. Jang E, Gu S, Poole B. Categorical reparameterization with Gumbel-Softmax. In: *International Conference on Learning Representations*, Toulon, France: ICLR publisher, 2017.
35. Zhang H, Zhang C, Li Z. *et al*. A new method of RNA secondary structure prediction based on convolutional neural network

and dynamic programming. *Front Genet* 2019;**10**:467. https://doi.org/10.3389/fgene.2019.00467.
36. Tan Z, Yinghan F, Sharma G. *et al*. TurboFold II: RNA structural alignment and secondary structure prediction informed by multiple homologs. *Nucleic Acids Res* 2017;**45**:11570. https://doi.org/10.1093/nar/gkx815.
37. Sloma MF, Mathews DH. Exact calculation of loop formation probability identifies folding motifs in RNA secondary structures. *RNA* 2016;**22**:1808–18. https://doi.org/10.1261/rna.053694.115.
38. Danaee P, Rouches M, Wiley M. *et al*. BpRNA: large-scale automated annotation and analysis of RNA secondary structure. *Nucleic Acids Res* 2018;**46**:5381–94. https://doi.org/10.1093/nar/gky285.
39. Kalvari I, Argasinska J, Quinones-Olvera N. *et al*. Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families. *Nucleic Acids Res* 2018;**46**:D335–42. https://doi.org/10.1093/nar/gkx1038.
40. Weinberg Z, Lünse CE, Corbino KA. *et al*. Detection of 224 candidate structured RNAs by comparative analysis of specific subsets of intergenic regions. *Nucleic Acids Res* 2017;**45**:10811–23. https://doi.org/10.1093/nar/gkx699.
41. Rose PW, Prlić A, Altunkaya A. *et al*. The RCSB protein data bank: integrative view of protein, gene and 3D structural information. *Nucleic Acids Res* 2017;**45**:D271–D281.
42. Limin F, Niu B, Zhu Z. *et al*. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* 2012;**28**:3150–2.
43. Do CB, Woods DA, Batzoglou S. Contrafold: RNA secondary structure prediction without physics-based models. *Bioinformatics* 2006;**22**:e90–8. https://doi.org/10.1093/bioinformatics/btl246.
44. Mao K, Wang J, Xiao Y. Prediction of RNA secondary structure with pseudoknots using coupled deep neural networks. *Biophys Rep* 2020;**6**:146–54. https://doi.org/10.1007/s41048-020-00114-x.
45. Paszke A, Gross S, Massa F. *et al*. Pytorch: an imperative style, high-performance deep learning library. *Adv Neural Inf Process Syst* 2019;**32**:8024–8035.
46. Kingma DP, Ba J. Adam: a method for stochastic optimization. In: *International Conference on Learning Representations*, San Diego, USA: ICLR publisher, 2015.
47. Zuker M. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res* 2003;**31**:3406–15. https://doi.org/10.1093/nar/gkg595.
48. Lorenz R, Bernhart SH, Höner C. *et al*. ViennaRNA package 2.0. *Algorithms Mol Biol* 2011;**6**:1–14. https://doi.org/10.1186/1748-7188-6-26.
49. Zakov S, Goldberg Y, Elhadad M. *et al*. Rich parameterization improves RNA structure prediction. *J Comput Biol* 2011;**18**:1525–42. https://doi.org/10.1089/cmb.2011.0184.
50. Nawrocki EP, Burge SW, Bateman A. *et al*. Rfam 12.0: updates to the RNA families database. *Nucleic Acids Res* 2015;**43**:D130–7. https://doi.org/10.1093/nar/gku1063.
51. Altschul SF, Madden TL, Schäffer AA. *et al*. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 1997;**25**:3389–402. https://doi.org/10.1093/nar/25.17.3389.
52. Nawrocki EP, Eddy SR. Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics* 2013;**29**:2933–5. https://doi.org/10.1093/bioinformatics/btt509.
53. Darty K, Denise A, Ponty Y. VARNA: interactive drawing and editing of the RNA secondary structure. *Bioinformatics* 2009;**25**:1974–5. https://doi.org/10.1093/bioinformatics/btp250.
54. Buitrón MG, Cahui RRT, Ríos EG. *et al*. CoDNaS-RNA: a database of conformational diversity in the native state of RNA. *Bioinformatics* 2022;**38**:1745–8. https://doi.org/10.1093/bioinformatics/btab858.

Briefings in Bioinformatics, 2025, **26(1)**, bbae618
https://doi.org/10.1093/bib/bbae618
Problem Solving Protocol