PLOS ONE

RESEARCH ARTICLE

De novo prediction of RNA 3D structures with deep generative models

Julius Ramakers[1☯]*, **Christopher Frederik Blum**[1☯], **Sabrina König**[1☯], **Stefan Harmeling**[2], **Markus Kollmann**[1]*

1 Department of Computer Science, Heinrich-Heine-Universität Düsseldorf, Düsseldorf, Germany,
2 Department of Computer Science, Technical University Dortmund, Dortmund, Germany

☯ These authors contributed equally to this work.
* ramakers@hhu.de (JR); markus.kollmann@hhu.de (MK)

OPEN ACCESS

Citation: Ramakers J, Blum CF, König S, Harmeling S, Kollmann M (2024) De novo prediction of RNA 3D structures with deep generative models. PLoS ONE 19(2): e0297105. https://doi.org/10.1371/journal.pone.0297105

Editor: Yang Zhang, University of Michigan, UNITED STATES

Received: May 31, 2023

Accepted: December 24, 2023

Published: February 15, 2024

Data Availability Statement: All code is available under the following public repository: https://github.com/ramakers/deep-rnafold All data is available under the following public repository. Here the mini-mal data underlying the results is placed in as well: https://uni-duesseldorf.sciebo.de/s/UapNNvwkCIVMHib The data repository contains also the training data and hence is big in size, so that we required our internal university repository. We are happy to share data also in other repositories if there are some available by PLOS that support 15GB capacity.

Abstract

We present a Deep Learning approach to predict 3D folding structures of RNAs from their nucleic acid sequence. Our approach combines an autoregressive Deep Generative Model, Monte Carlo Tree Search, and a score model to find and rank the most likely folding structures for a given RNA sequence. We show that RNA *de novo* structure prediction by deep learning is possible at atom resolution, despite the low number of experimentally measured structures that can be used for training. We confirm the predictive power of our approach by achieving competitive results in a retrospective evaluation of the RNA-Puzzles prediction challenges, without using structural contact information from multiple sequence alignments or additional data from chemical probing experiments. Blind predictions for recent RNA-Puzzle challenges under the name "Dfold" further support the competitive performance of our approach.

Introduction

Ribonucleic acids (RNAs) are polymeric molecules that can act as information messengers, mediators, and regulators in the expression of genes. The specific function of RNA is tightly associated with the 3D folding structure, which in turn is determined by its sequence of nucleobases. The accurate prediction of RNA 3D structure from its primary sequence would advance the design of synthetic RNA for biotechnological or therapeutic purposes and help to improve RNA vaccines or RNA based gene therapies. Using Deep Generative Models for RNA structure prediction circumvents the complex tasks of formulating an energy function from which structural candidates can be generated but requires a sufficient amount of examples to learn the complex conformational states RNA molecules can take. The functional diversity of RNA in living cells is a consequence of its ability to form specific three-dimensional (3D) folding structures that allow for interaction with DNA, RNA, proteins, and small molecules [1, 2]. Understanding the relationships between sequence, structure, and function of RNA is essential for understanding the function of living cells and particularly useful for the design of RNA therapeutics [3, 4]. Furthermore, the automated prediction and the targeted design of RNA tertiary structure would be an important step to further improve RNA therapeutics and to advance the field of RNA biotechnology in general [5].

Funding: JR and CB acknowledge funding from the Start-up Transfer.NRW (EFFRE-0400380, Nordrhein-Westfalen). CB MK and SH acknowledge funding from the Manchot Foundation, supporting interdisciplinary Artificial Intelligence research at the Heinrich Heine University. Both funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Algorithms for predicting RNA 3D structure from nucleotide sequence [6] are dominated by four approaches: (i) template based methods such as FARFAR2 [7, 8] and 3dRNA2 [9, 10], which decompose known structures into 1- to 3-mer fragments and combinatorially reassemble them to find the structures with lowest molecular interaction energies [11], (ii) coarse grained force field methods that minimise interaction energy by stochastically displacing groups of atoms like SimRNA and RNA-BRiQ [12, 13], (iii) comparative modelling methods that are based on the availability of homologous structures, and (iv) machine learning approaches [14, 15] that combine sequence and chemical probing information to generate candidate structures. Despite the steady increase in affordable computing power and the use of more accurate energy functions [11, 16], the *de novo* structure prediction of larger RNAs (>80nt) still remains challenging [7]. For secondary structure prediction deep learning based models like RNA-FM, U-Fold and SPOT-RNA have allready surpassed shallow networks and energy based methods [17–19]. Hence, it is appealing to study deep learning in the tertiary prediction setup and recent studies with DRFold have shown that end-to-end deep learning for tertiary structure prediction can be achieved [20]. Historically, the prediction challenges for RNA structure prediction algorithms started with benchmarks for the prediction of small scale structures [21] (2012) up to larger structures in [22] (2015), followed by more complex folds such as riboswitches and ribozymes [23] (2017). In this historic context fragment-assembly methods perform best, giving especially leading structure predictions for larger sequences.

For proteins, the benchmark for predicting 3D structures with atomic resolution is set by deep learning approaches that take sequence information as input and predict both the distances between C_α or C_β atoms, the dihedral angles of the polypeptide backbone, and the conformation of the side chains [24, 25]. Here, the accurate prediction of the global folding structure crucially depends on the existence of a sufficient amount of homologous sequences from multiple sequence alignments (MSA), which allows to identify at least some of the residues that are in contact [26]. These global structural constraints can be inferred from correlations between amino acid substitution frequencies that arise from an evolutionary selection pressure for stably folded protein structures [27]. The ability of deep neural networks to identify and generate complex statistical patterns in high dimensional spaces and to generalise well across training examples makes deep learning approaches conceptually attractive for predicting protein structures. Hence, within the scientific community concerned with RNA, the question on when deep learning will lead to breakthrough has allready been raised [28] However, deep learning approaches are in general data hungry and structure predictions strongly benefits from a sufficiently large number of homologous sequences for each high resolution structural example in the training set. For the same reason, the use of clever data augmentation strategies are crucial to achieve good performance [29].

Structure prediction for RNA shows some fundamental differences to proteins. First, in contrast to almost all proteins, RNAs often fold into different alternative structures under physiological conditions that are either stable or visited over time with high probability [30]. Second, for RNA the conformation of the phosphate backbone is strongly constraint by the pairing of nucleobases, whereas for proteins the spatial location of the side chains is strongly constraint by the polypeptid backbone. This difference arises from the fact that the secondary structure of proteins is determined by hydrogen bonds within the peptide backbone, whereas the secondary structure of RNA is determined by hydrogen bonds between nucleobases. Third, as training of large deep learning models requires a large amount of independent training examples, the two orders of magnitude less available structures for RNAs in comparison to proteins implies stronger restrictions on the model complexity for RNA structure prediction. Finally, the less conserved RNA structures make it much harder to identify homologs for MSA and therefore the crucial information about global folding constraints is in many cases not

accessible. On the contrary, there exist structural probing methods to estimate the probability of each nucleotide to be part a of a base paring interaction, such as SHAPE [31] or DMS [32]. However, unlike MSAs, structural probing methods can only give an estimate if a nucleotide is in contact, but lack direct information about the contact partner. Moreover, structural probing data represents an ensemble average over the structural conformations that a given RNA can take and therefore can provide only useful information if the secondary structure is sufficiently stable.

Materials and methods

To encode 3D RNA structures, we used a rotational invariant representation that was given by the Euclidean distances between nucleotides, with each nucleotide position uniquely determined by a set of 5 selected atoms, where different sets were taken for purines and pyrimidines (Fig 1). We made use of a Vector Quantised Variational Autoencoder [33] (VQ-VAE) to compress the 5×5 Euclidean distances between the selected atoms into K classes for each possible nucleotide pair. We refer to these classes as distance classes, as the $K = 3$ classes we used throughout this work agree well with the qualitative distance measures "near", "intermediate", and "far" (Supplementary Information). The encoded distance classes represent the targets

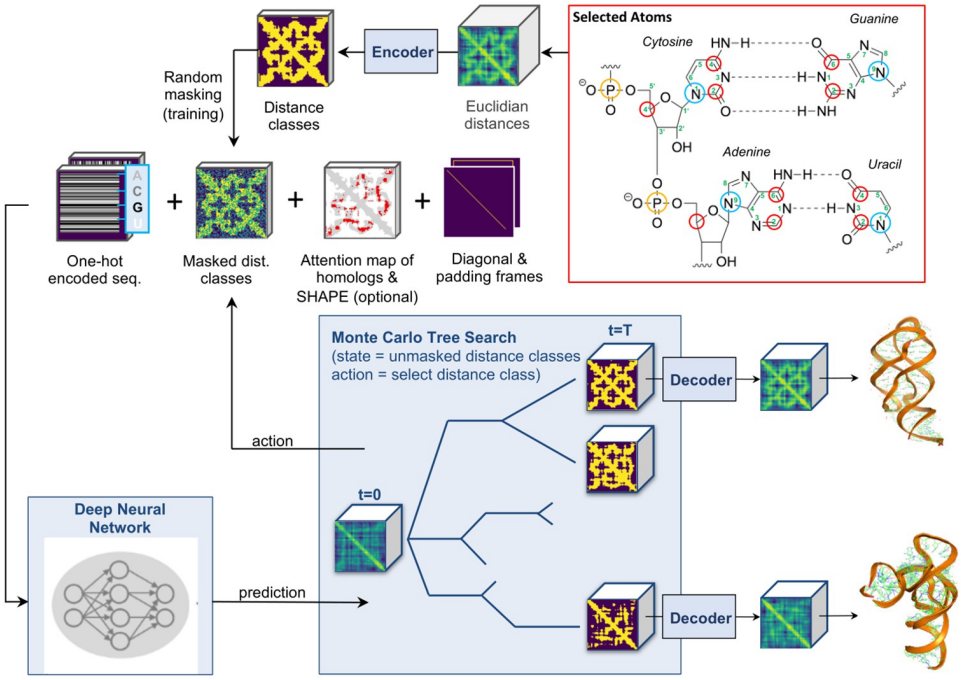

Fig 1. Data flowchart for the RNA structure generation process. The PDB structure is represented by Euclidean distances between nucleotide pairs and there are eight atom types for the RNA nucleotides, see also Table 1. The position of each different nucleotide is determined by five out of eight selected atoms, resulting in an 8×8 matrix for each single nucleotide which can be flattened into a vector. The resulting $L \times L \times 64$ Euclidean distances for all nucleotide pairs are encoded into $L \times L \times K$ discrete distance classes by a VQ-VAE [33], we choose $K = 3$. The generation process uses a Deep Neural Network (DNN) to predict probability values for the distances classes of shape $L \times L \times K$, so that for each pixel a via softmax a probabilitiy distribution of the K distance classes is learned. From these predictions a single distance class for a single nucleotide pair is selected according to the MCTS policy (Methods) to iteratively generate a path in the search tree. At each iteration the currently selected distance classes and the sequence information are presented as input to the DNN. Once all distance classes are selected, the Euclidean distances can be recovered by the VQ-VAE decoder. A Score Model (Methods) selects the most promising generated structures, which are then further refined by minimising a coarse grained molecular energy function [13].

https://doi.org/10.1371/journal.pone.0297105.g001

used for training a Deep Generative Model that takes sequence information and masked targets as input. The task of the Generative Model was to predict the probabilities of the masked distance classes. For the masking, we first selected the fraction of nucleotide pairs (pixels) to be masked by randomly drawing an integer number n from the set $\{1, 2,.., (L \times (L-1))^2/2\}$, with L the sequence length, and then randomly selecting n out of $(L \times (L-1))^2/2$ pixels whose one-hot encoded target values were then overwritten by assigning each distance class the same value. We only need to select up to $(L \times (L-1))^2/2$ pixels since distance matrices are symmetric. Training neural network architectures with masked targets on input shows surprisingly strong generalisation behaviour and has resulted in state-of-the-art results for learning words representation in Natural Language Processing (NLP) and for image generation in computer vision [34–37]. After training, a structure can be iteratively built up by sampling a distance class for each nucleotide pair according to a MCTS search algorithm (Methods) and presenting the selected distance class at the input (Fig 1). Although our generative model allows to estimate the likelihood for each predicted structure by making use of the chain rule for probability mass functions [36], this value is in general unreliable [38]. We therefore trained a Score Model (Methods) that allowed to score the match between sequence and generated structures, similar to a value function in reinforcement learning [39]. Each predicted, one-hot encoded distance matrix with high score was mapped back to an Euclidian distance matrix, using the decoder of the VQ-VAE. The Euclidean distances were further fine tuned by minimising a coarse grained, physical RNA energy function [13].

In our approach we included some best practices for training deep neural networks. First, deep neural networks strongly benefit from end-to-end learning, where gradients for updating parameters are allowed to propagate from the objective function back to the input, thereby avoiding extensive preprocessing steps that might reduce the information content [40]. Second, the inductive bias induced by the network architecture should match the structure of the data. We therefore combined self-attention layers (Supplementary Information) to extract long-range interactions within the RNA sequence and used convolutional layers to predict local correlations in the RNA 3D structure [41, 42]. Third, the final performance of a deep learning model depends significantly on (i) the neural network size, (ii) the amount of training data, and (iii) the training time. The generic empirical observation, which is also confirmed in this work, is that increasing (i)-(iii) increases the prediction accuracy [43]. Consequently, we employed advanced data augmentation techniques, which allowed us to train larger networks that were able to model more complex mappings and achieve better generalisation.

Data extraction and preprocessing

For the construction of our training, validation and test set we extracted 2581 RNA molecule entries from the RNAsolo database [44] with a resolution of less than 4 Å. From that, we distilled out structures that are non-redundant RNA only folds and split large complex structures into their single chain components. We discarded RNA structures in complex with protein/DNA and multichain RNAs and removed sequences with non A,G,C,U content. By including RNA structures of the RNA-Puzzles challenges from the PDB that were missing in RNAsolo, we arrived at 1454 RNA single chain structures. The extracted structures were grouped according to their Bowling Green State University (BGSU) RNA class membership [45], on which we performed hierarchical clustering based on sequence similarity, as high sequence similarity typically implies high structural similarity. Clusters were build based on a sequence similarity cutoff of 0.7. The test set was build from clusters that comprise the RNA-Puzzles. The complete dataset was split on cluster level into training, validation, and test set, with cardinality 1127, 327, and 78, respectively. We perform such a stringent splitting to detect overfitting, as larger

deep learning models can memorise structures based on sequence input. To augment the structural data, we carried out Molecular Dynamics (MD) simulations [13] for each of the 1127 sequences that were initialised by the atom positions of structural variants that correspond to the same PDB id (NMR ensembles or symmetrical copies of biological assemblies). We ran the simulations independently 5 times with a varying number of time steps to obtain "drifted structures" with about 3 Å root-mean-square error (RMSE) to the original PDB structure. The drifted structures induce noise on training targets, similar to label smoothing [46, 47], which is a regularisation technique that has been introduced to avoid overconfident predictions. From the 1454 structures, 467 had a sequence length $L \geq 100$nt. We used the sequences of length $L \geq 100$nt to generated additional structure-sequence pairs by randomly cropping them to length $L = 100$nt. For each cropped structure, we memorised the *contact nucleotides*, which are those nucleotides of the crop that have less than 3.3 Å distance to the remaining nucleotides of the original structure. We included only crops in the training dataset with less than 5% contact nucleotides. This very stringent cutoff reduces the bias of training examples towards contact constrained folding structures. We used a binary indicator variables to mark all possible pairs of contact nucleotides and showed the corresponding distance classes for these pairs as fixed input during training. As cropped structures made up most of the training set, the model effectively learned to predict substructures that were constrained by the remaining part of the RNA structure. To predict free folding RNA structures we take the trained prediction model for the cropped structures and set the binary indicator variables to zero. This data augmentation approach is similar to the concept of non-leaking data set augmentations [48]. As our dataset contains large structures with length up to 1513 nucleotides, random cropping results in strong data augmentation with a total of 6245 unique structures. After generating drifted structures for each unique structure, the augmented training, validation, and test sets comprise 27644, 3270, and 78 structures, respectively. We determined the position of each nucleotide by 5 selected atoms (Fig 1) and compressed the 25 possible real distances between the selected atoms for any nucleotide pair into $K = 3$ distance classes using a Vector Quantised Variational Autoencoder (VQ-VAE).

Autoregressive generative model

The generation of a 3D structure, $\boldsymbol{s}$, from sequence information, $\boldsymbol{x}$, was carried out iteratively by first selecting a nucleotide pair (pixel) with index $i \in \{1,..,N\}$ from the $N = L(L-1)/2$ possible pairings and subsequently selecting a distance class $k_i \in \{1,..,K\}$ according to the class probabilities predicted by the Generative Model, $P(\boldsymbol{k}|\boldsymbol{s}_t, x)$. The selected distance class was then one-hot encoded, resulting in an updated input structure $\boldsymbol{s}_{t+1} \leftarrow \boldsymbol{s}_t$. We denote by $a_t = k_{i_t}$ the "action" for the t-th iterative step in the generation process and defined the current structural state by $\boldsymbol{s}_t = (a_t, \ldots, a_1)$. Actions are never overwritten during the generation process, which starts from an empty set of actions $\boldsymbol{s}_0$ by "masking" all pixels as defined below. The Generative Model was realised by a feed-forward neural network $P(\boldsymbol{k}|\boldsymbol{s}_t, x) = \prod_i P_i(k_i|\boldsymbol{s}_t, x)$ that predicted probabilities for all distance classes $\boldsymbol{k} = (k_1, k_2, \ldots, k_N)$ in parallel, given the previous actions $\boldsymbol{s}_t$. The conditional independence of the predicted class probabilities is a consequence of the deterministic network architecture, where the outputs (class probabilities for each pixel) are uniquely determined by the input. The complete input feature map of the autoregressive model comprised of (i) a one-hot encoding of the 16 possible nucleotide pairs (AA,AC, . . ., UU) for each pixel, (ii) the already set distance classes in the autoregressive process, $\boldsymbol{s}_t$, (iii) coordinate frames that included the diagonal as symmetry axis and padded regions for sequences of length $L < 100$, (iv) the output of a Self-Attention layer [42] that takes structural probing data and homologous sequences as input (Fig 1). Training was carried out by showing

for $\boldsymbol{s}_t$ a random fraction of distance classes (masked target) at the input, where the one-hot encoding for the K distances classes was substituted by information that was related to the target logits, e.g. (1, −1, −1) if the target was the first distance class and (0, 0, 0) if the target class was masked. The number of masked distance classes shown at the input was distributed according to a truncated half-normal distribution, to enforce that almost complete targets are shown less frequently during training. The feed-forward network architecture was a residual network with 16 residual blocks and 26 channels in each hidden layer, trained by early stopping (Supplementary Information). The generative model allowed to compute likelihood estimates of a structure $\boldsymbol{s}_N$ for a given nucleotide sequence $\boldsymbol{x}$ by making use of the chain rule of probability, $P(s_N|x) = \prod_{t=1}^{N} P(a_t|\boldsymbol{s}_{t-1}, x)$.

Score model

The likelihood estimate was improved by learning a Score Model $D(\boldsymbol{s}_N, \boldsymbol{s}'_N; \boldsymbol{x})$ (discriminator) that was trained to distinguish between correct and incorrect distance-map/sequence pairs by maximising the objective

$$\begin{aligned} J(D) &= \mathbb{E}_{\boldsymbol{s}_N \sim P_{true}(\boldsymbol{s}_N, \boldsymbol{x})} \mathbb{E}_{\boldsymbol{s}'_N \sim P_{false}(\boldsymbol{s}'_N, \boldsymbol{x}))} \Big[\log D(\boldsymbol{s}_N, \boldsymbol{s}'_N; \boldsymbol{x}) \Big] \\ &+ \mathbb{E}_{\boldsymbol{s}_N \sim P_{true}(\boldsymbol{s}_N, \boldsymbol{x})} \mathbb{E}_{\boldsymbol{s}'_N \sim P_{false}(\boldsymbol{s}'_N, \boldsymbol{x})} \Big[\log(1 - D(\boldsymbol{s}'_N, \boldsymbol{s}_N; \boldsymbol{x})) \Big] \end{aligned}$$

with D defined by

$$D^*(\boldsymbol{s}_N, \boldsymbol{s}'_N; \boldsymbol{x}) = \frac{1}{1 + \exp[f(\boldsymbol{s}'_N, \boldsymbol{x}) - f(\boldsymbol{s}_N, \boldsymbol{x})]}$$

and $f(\boldsymbol{s}'_N, \boldsymbol{x})$ being the scalar output of a deep neural network. The theoretically optimal solution $f(\boldsymbol{s}_N, \boldsymbol{x}) = \log \frac{P_{true}(\boldsymbol{s}_N, \boldsymbol{x})}{P_{false}(\boldsymbol{s}_N, \boldsymbol{x})}$ is typically not reached by optimisers based on stochastic gradient decent [49]. Here, "true" corresponds to original PDB examples and "false" to PDB examples with drifted atom position using Molecular Dynamics Simulations [13] under high temperature and encoded by the VQ-VAE or distance-maps predicted from the Generative Model. The discriminator compares two complete distance-maps with respect to their match to a given sequence, which is in contrast to the absolute likelihood estimate from the chain rule of probability. The value $f(\boldsymbol{s}'_N, \boldsymbol{x})$ is used to rank the predictions that are sampled from the Generative Model. For the Score Model, we used a Residual Network architecture with 8 residual blocks, where blocks were connected by down-sampling layers using stride 2 convolutions (Supplementary Information).

Structural sampling

Unlike proteins, RNAs frequently fold into different structures under physiological conditions. To identify the structures that occur with high probability, we had to sample the large combinatorial space of possible distance-maps and rank them according to their corresponding likelihood. For RNAs of length L = 100 nucleotides, the combinatorial space of allowed distances is given by $K^{L(L-1)/2} > 10^{2000}$ for $K \geq 3$ and thus exceeds the number of possible games, $\sim 10^{700}$, that can played in the board game Go. To realise fast sampling, we borrowed search strategies from reinforcement learning (RL). We aimed to find the best sequential ordering (a_t, . . ., a_1) for presenting distance classes at the input, such that after a minimum number of autoregressive steps the probability masses for the remaining nucleotide distances became highly concentrated into one class. This ordering allowed to predict the final distance map after $T \ll N$ steps by selecting the most likely distance classes for the set of remaining masked pixels, M_T,

in parallel

$$\{k_j\}_{j\in M_T} = \arg\max \prod_{j\in M_T} P(k_j|\boldsymbol{s}_T, x)$$

To find a close to optimal autoregressive ordering, we developed a variant of the Monte Carlo Tree Search (MCTS) that accounts for the fact that some actions affect the global distance-map and hence structure more that others. For a given nucleotide sequence, a tree of possible RNA distance-maps can be built iteratively by connecting incomplete distance-maps (nodes), s_t, with actions, a_t, such that each leaf node s_L of the current tree can be reached by a unique path of actions $s_L = (a_L, \ldots, a_1)$. For selecting the actions to reach a leaf node from the root node (empty set of actions), s_0, we followed the selection rule (policy) [50]

$$a_{t+1} = \underset{a}{\text{argmax}}(Q(s_t, a) + U(s_t, a)) \quad ; \quad U(s_t, a) = c_p \frac{\sqrt{\sum_a N(s_t, a)}}{1 + N(s_t, a)}$$

with $Q(s_t, a)$ the expected entropy reduction of $P(a|\boldsymbol{s}_t, x)$ if action a is taken, $N(s_t, a)$ a counter how often actions that connect the root node with a leaf node pass through the state-action pair (s_t, a) under the actual policy, and $U(s_t, a)$ a term that upweights rarely visited actions (exploration), with c_p being a tuneable constant. After reaching a leaf node, s_L, the tree is expanded by randomly selecting a subset S_R of the remaining masked pixels, and from S_R a subset $S_H \subset S_R$ of actions that result in sufficiently strong entropy reduction $\Delta H_L < \lambda \log K$ for the predicted class probabilities, with $\lambda = 1$ and

$$\begin{aligned} \Delta H_L &= H(\boldsymbol{s}_{L+1}, \boldsymbol{x}) - H(\boldsymbol{s}_L, \boldsymbol{x}) \\ H(\boldsymbol{s}, \boldsymbol{x}) &= -\sum_{j=1}^{N}\sum_{k_j=1}^{K} P(k_j|\boldsymbol{s}, x) \log P(k_j|\boldsymbol{s}, x) \end{aligned}$$

For $|S_H| = \emptyset$ the leaf node s_L is added to set of terminal nodes $S_T \leftarrow S_T \cup s_L$ and for $|S_H| \neq \emptyset$ the tree is enlarged by $|S_H|$ nodes in parallel, initialising $N(s_L, a) = 1$ and $Q(s_L, a) = v(\boldsymbol{s}_{L+1}|\boldsymbol{x})$ for all $a \in S_H$, with value function the entropy reduction rate

$$v(\boldsymbol{s}_L|\boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T} \frac{H(\boldsymbol{s}_0|\boldsymbol{x}) - H(\boldsymbol{s}_L|\boldsymbol{x})}{H(\boldsymbol{s}_0|\boldsymbol{x})}$$

Along the path of actions from the root node to $s_L = (a_1, .., a_L)$, we updated for all $t \in \{1, .., L\}$ the visit count $N(s_t, a_t) \leftarrow N(s_t, a_t) + |S|$ and subsequently the expected reward $Q(s_t, a_t|\boldsymbol{x}) \leftarrow Q(s_t, a_t|\boldsymbol{x}) + \frac{|S_H|}{N(s_t, a_t)}(v(\boldsymbol{s}^*_{L+1}|\boldsymbol{x}) - Q(s_t, a_t|\boldsymbol{x}))$, using the 'winner takes it all' value function $v(\boldsymbol{s}^*_L|\boldsymbol{x})$, with $s^*_L = \text{argmax}_{a_L \in S_H} v(\boldsymbol{s}_L|\boldsymbol{x})$. For leaf nodes that were terminal nodes, $s_L \in S_T$, we updated $N(s_t, a_t) \leftarrow N(s_t, a_t) + N_{expl}$, with $N_{expl} = 10$ a hyperparameter, along the path and leave $Q(s_t, a_t)$ unchanged to enforce exploration of different distance-maps and thus structures.

Structural ensemble

We generated complete distance-maps for the set of terminal leaf nodes S_T by argmax sampling from $P(\boldsymbol{k}|\boldsymbol{s}_T, x)$, as introduced above. We ranked the complete distance-map relative to each other according to the discriminator output. The resulting ensemble was a subset of the possible distance-maps that an RNA can take. As we started out to the find the most likely distance-map by MCTS, with alternative distance-maps as a by-product of the search, the resulting ensemble was highly biased towards distance-maps with high likelihood.

Refinement

For the RNA-Puzzles challenges (Fig 2) we carried out refinement steps using a coarse grained force field method. For the sampled structural ensemble we computed 1000 simulation runs using SimRNA. These simulation runs typically cluster near local optima. We used the built-in SimRNA clustering function based on all Molecular Dynamics traces with an energy cut off at 25% and computed five spectral clusters based on 5.0, 10.0, 15.0, 20.0 and 25.0 Å pairwise distance between clusters. The cluster centers represent candidate structures from which the one with highest likelihood was chosen for the retrospective evaluation and all 5 candidates were submitted to the blind RNA-Puzzless prediction challenge (Supplementary Material).

Results and discussion

We first tested the reconstruction accuracy of the VQ-VAE as a function of the number of distance classes (Fig 2a). For $K = 8$ classes, the reconstruction error approached the average experimental resolution of 2.8 Å root-mean-square error (RMSE). For $K = 3$ classes the median reconstruction error was still in the range of the best predictions with 4 Å RMSE. Next, we

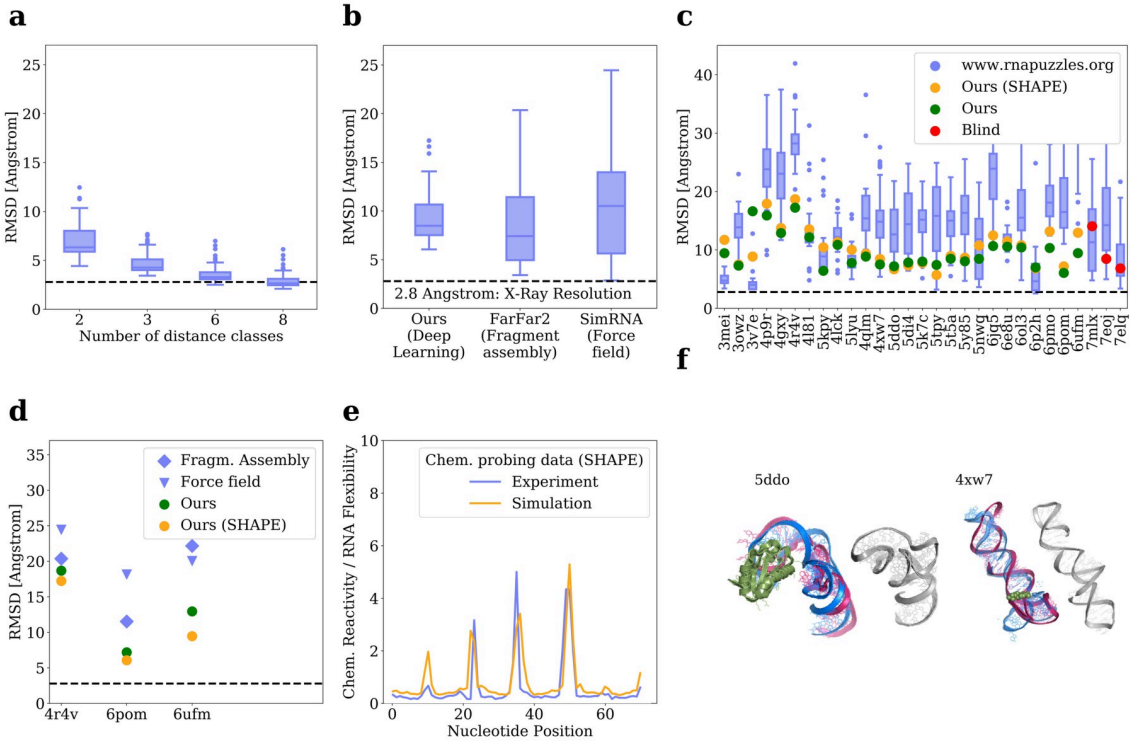

Fig 2. Results: Evaluation of the structural predictions. A, Reconstruction error resulting from encoding and decoding RNA 3D structures of the test set as a function of the number of distance classes, **B,C** Simulated blind tests of the RNA-Puzzles Challenges in comparison to other approaches, with or without simulated SHAPE reactivity data and homologous sequences as additional input. The three right most predictions are real blind submission from the latest puzzle round 33 (7mlx, 7eoj, 7elq). Puzzles 6ufm, 6pom, and 6pmo are large compounds of tRNA-Riboswitch complexes. Puzzle predictions are sorted in descending order of the sequence length (left to right) **D,** Reconstruction error of longer RNA-Puzzles that lack both structural and sequence homology (PDB 4R4V: 185 nt,) in comparison to (PDB 4QLM: 108 nt, PDB 6pom tBox: 75, PDB 6ufm Complex: 175 nt) for which structural and sequence homologs are available. Structures of length >175 nt were cut into substructure of length ≤100 nt, aligned with PyMOL, and RMSD calculated as average over all substructures. **E,** Simulated chemical probing data in comparison with experimentally measured reactivities [52] (SHAPE) for PDB 1Y26, **F,** Most likely alternative structures, as predicted by the Score Model, for the Gluatamine riboswitch and the ZMP riboswitch (S3 Fig).

https://doi.org/10.1371/journal.pone.0297105.g002

simulated a blind test by evaluating the accuracy on a held out test set that is given by the crystal structures of the RNA-Puzzles challenges [51] (Fig 2b and 2c), including only free folding RNA structures.

We observed significant improvements for RNA structure prediction problems that were classified as difficult [7] (RNA-Puzzles 7, 27, and 28) due to their longer sequence (>100nt) and their lack of homology to known structures (e.g. 6ufm in Fig 2d, S3 Fig). The difficulty can be seen that all competing puzzle submissions have a high RMSD on the choosen examples, which is also true for our approach. We have also participated in three blind predictions (7mlx, 7eoj, 7elq) under the submission name "Dfold" with single digit RMSD reconstruction error, which is in line with our retrospective evaluation. The blind predictions confirm that our deep learning based approach can indeed predict new structures and is not only memorizing the training data.

To investigate the effect of structural probing data (SHAPE), we used a force field model [13] to simulate SHAPE reactivities which are shown at the input during training. We thereby assumed that single stranded RNA is more flexible than double stranded RNA and thus shows higher mean squared displacement (MSD) of atoms during the force field simulations. The simulated MSD values show good agreement with the experimentally determined SHAPE reactivities (Fig 2e). We observed a small improvement in prediction accuracy on average when we presented both SHAPE data and MSAs of homologous sequences at the input, which indicates that the additional constraints imposed by simulated SHAPE data and evolutionarily constrained nucleotide-nucleotide interactions provide only little additional information to our model for RNAs of length $L \leq 100$nt. However, we found that this additional input data allowed to infer global structural information (S2 Fig), thereby accelerating MCTS. This acceleration might become crucial in cases where exploring the global structural space by MCTS is the limiting factor.

The ability of our approach to find alternative structures can be used to predict the different states of riboswitches (Fig 2f). We simulated a blind prediction test by removing all homologous structures from the training set for the Glutamine Riboswitch (PDB: 5*DDO*) and the ZMP Ribosowitch (PDB: 4*XW*7). The predicted alternative structures, which represent two highest ranked branches of the MCTS by the Score Model, confirm the general viewpoint that riboswitches work by a ligand mediated stabilisation of one structural conformation.

Despite the strong advances in protein structure prediction using Deep Learning approaches, RNA 3D structure prediction remains challenging for longer sequences. The reason can be attributed to the limited number of experimentally determined structures in public databases ($\sim 10^2$ less structures for RNAs as for proteins) and the fact that some RNAs can fold in different structural variants under physiological conditions. The existence of an ensemble of possible RNA structures cannot be appropriately addressed by deterministic feed-forward network architectures within deep learning approaches that assign each sequence exactly one structure [24]. We therefore employed a structural sampling approach that combines a deep generative model with an efficient search method through structural space. Our approach provides a more efficient way of generating structural candidates than fragment assembly approaches. This higher sampling efficiency may become crucial for longer RNA sequences, where sampling the combinatorial space using structural elements becomes computationally prohibitive. On the downside, predictions by deep neural networks frequently violate physical constraints and require additional relaxation steps to generate valid structures with atom resolution [24]. For that, it is worth exploring additional, e.g. stereochemistry data to stack next to the SHAPE matrix for an uplift in prediction performance as validated in [53]. We therefore expect that combining more RNA specific datasets and the use of an advanced score model with atom resolution, such as ARES [16], to relax the outcome of a deep generative model is a

promising strategy for further improvements of RNA 3D structure prediction. We have also tested our model under various data splits and need to highlight that without a rigorous training, validation and test set split, deep learning based models are highly exposed to overfitting, especially given the small amount of experimentally determined RNA 3D structures available.

Supplementary material

Data preparation and preprocessing

Source data. For the construction of our training, validation and test set we extracted 2581 RNA molecule entries from the RNAsolo database [44] with a resolution of less than 4 Å. From that, we distilled out structures that are RNA only folds and we splitted large complex structures into their single chain components. We further added additional RNAs that are not availaible in RNAsolo directly from the PDB (especially from the RNApuzzles) and cleaned the so derived raw dataset: In particular, we only kept a structure if (i) it had only RNA chains (entry-type in pdb_entry_type.txt was "nuc"), (ii) the resolution method was either X-ray diffraction or NMR ("method" in pdb_entry_type.txt was either "diffration" or "NMR"), (iii) the PDB structure was downloadable from http://files.rcsb.org/download/, (iv) the "resNames" were either A, C, G or U, (v) the chain had at least 14 nucleotides, or at least 7 nucleotides if the structure was composed of multiple chains, (vi) for structures with 3 or more chains, all chains were given by unique sequences.

Thus, we derived with 1454 RNA single chain structures. The so extracted structures were grouped according to their BGSU class members [45] to not have same BGSU class members from the training set in the validation and test set. To further enforce a structural difference between the training and validation and test set, we enforce a splitting with an analysis of sequence similarity, using hierarchical clustering with a similarity cutoff of 0.7. We split the clusters into training, validation, and test sets (1127, 327, and 78 respectively). All RNA-Puzzles are kept in the test set to have a represantative blind prediction simulation. The validation set is derived from all class members and clusters of similar sequence from the puzzle test set so that no class members from BGSU as well as class members of similar sequence are in the training set. The training set only contains pdb entries, that have no class and similarity member in the validation and test set and hence, can be considered as structural different, so that the networks can not memorize the structures. If we do not perform such a stringent splitting we experience much stronger benchmark results, pointing to the fact that deep learning models can just memorize structures based on sequence input.

Neural network input data format. All PDB structures were converted into a reduced 5-atom positional representation for each residue [13] (Fig 1). For Guanosine monophosphate and Adenosine monophosphates, we used the P, C4', C2, C6 and N9 atoms. For Cytosine monophosphates and Uridine monophosphates, we used the P, C4', C2, C4 and N1 atoms. To distinguish between purine and pyrimidine residues (G/A and C/U, respectively), we encoded the cartesian coordinates (x, y and z) of the 5 atoms by an 8×3 coordinate matrix and indicated the valid atoms (rows) by an 8×1 mask array, see Table 1.

When atoms were missing in a structure—which occurs frequently for the leading phosphate group—the corresponding coordinate and mask values were set to zero. The two atoms that determined the position of the phosphate backbone (first two rows in coordinate matrix) are shared between purines and pyrimidines. While both Purines and Pyrimidines have C2-atoms, this atom does not occur at the same position in the nucleobases' ring structures and hence was encoded separately (see red frame in Fig 1).

Distance and mask tensors. For an RNA sequence of length L, we computed for all of the $L \times L$ possible residue pairs the Euclidean distances between the encoded atoms. The resulting

Table 1. Encoding of purine (G/A) and pyrimidine (C/U) coordinates.

Atom type	Pyrimidines Matrix & Mask	Purine Matrix & Mask
P (phosphate)	$\begin{pmatrix} x_P & y_P & z_P \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix}$	$\begin{pmatrix} x_P & y_P & z_P \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix}$
C4′(sugar)	$\begin{pmatrix} x_{C4'} & y_{C4'} & z_{C4'} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix}$	$\begin{pmatrix} x_{C4'} & y_{C4'} & z_{C4'} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix}$
C2 (G/A only)	$\begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \end{pmatrix}$	$\begin{pmatrix} x_{C2} & y_{C2} & z_{C2} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix}$
C6 (G/A only)	$\begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \end{pmatrix}$	$\begin{pmatrix} x_{C6} & y_{C6} & z_{C6} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix}$
N9 (G/A only)	$\begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \end{pmatrix}$	$\begin{pmatrix} x_{N9} & y_{N9} & z_{N9} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix}$
C2 (C/U only)	$\begin{pmatrix} x_{C2} & y_{C2} & z_{C2} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \end{pmatrix}$
C4 (C/U only)	$\begin{pmatrix} x_{C4} & y_{C4} & z_{C4} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \end{pmatrix}$
N1 (C/U only)	$\begin{pmatrix} x_{N1} & y_{N1} & z_{N1} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \end{pmatrix}$

https://doi.org/10.1371/journal.pone.0297105.t001

$8L \times 8L$ distance matrix was re-shaped into a $L \times L \times 64$ distance Tensor D and symmetrized to satisfy the symmetry condition $D_{ijk} = D_{jik}$. For example, $D_{1L1} = D_{L11}$ is the Euclidian distance between the phosphate atoms of the first and the last residue.

Some atom-atom distances in this distance tensor D, however, did not correspond to any meaningful values because the corresponding atoms were missing and thus the coordinate entries in the 8×3 matrices were zero. The respective elements in D were set to zero by multiplying distance tensor D with a mask tensor M that was calculated as follows. First, the mask arrays of size 8×1 (Table 1) for the L residues were stacked to a single array of size $8L \times 1$ and the outer product with itself was calculated to obtain a matrix of size $8L \times 8L$. This matrix was then reshaped into a tensor of size $L \times L \times 64$ and symmetrized to obtain the mask tensor M.

Structures with long or multiple chains. When a structure had multiple chains or a chain's length exceeded 100 nt, we carried out the following steps to obtain multiple, smaller substructures suitable for model training. First, when a structure had multiple chains, a chain was randomly selected with probability proportional to its length and used as a substructure. Then, if that chain's length exceeded 100 residues, a random, continuous subsection of that chain was cut out and used as the substructure instead. For each residue in this substructure, the distances to the residues of the remaining, overall structure were calculated. For distances below a threshold of 3.3 Å, the corresponding residues of the substructure were flagged as "fixed" and the corresponding distance classes between "fixed" residues were presented at the input during training of the generator.

Data augmentation. SimRNA is a 3D RNA structure prediction software that makes use of coarse-grained residue representations and Molecular Dynamics Methods to sample the conformational space [13]. This program starts with a circular initial RNA structure (that resembles a snake biting it's own tail) and then folds the RNA to minimize an energy function while slowly cooling down the thermodynamic system. To do data augmentation, we used SimRNA the opposite way: we started with the original PDB structure and then increased the temperature to "drift away" from the original structure. Using this method, we generated 100 of such "drift structures" for each training example that were approximately 1, 3, 5 and 10 Å RMSE away from the original PDB structure.

Simulated SHAPE data. SHAPE (selective hydroxyl acylation analyzed by primer extension) [52] is a method for obtaining RNA secondary structure information. SHAPE exploits the fact that RNA residues that do not engage in base pairing, such as residues in dangling ends or loops, react more easily with certain reagents, making their detection possible by primer extension. In other words, higher SHAPE reactivities indicate regions of higher RNA flexibility. SHAPE data generated under comparable experimental conditions is only available

for a very limited number of PDB structures [52]. We hence used above mentioned drift data (see "Data augmentation") to simulate SHAPE data. Specifically, we structurally aligned original PDB structures and their drift structures and used the average absolute position-specific deviation (that is, the RMSE between individual atoms) as simulated SHAPE data.

Experimental SHAPE data. We used publicly available SHAPE data [52]. As the simulated SHAPE data only correlates with but does not exactly match the experimental SHAPE data, we rescaled the experimental SHAPE data. For this purpose, we mapped all data percentiles between experimental and simulated SHAPE data (for example, a SHAPE value that fell into percentile 10 among the original SHAPE data was mapped to the 10^{th} percentile of the simulated SHAPE data). This mapping was learned using the following PDB IDs: 2*L*1*V*, 2*K*95, 3*PDR*, 3*DIG*, 1*P*5*O*, 3*G*78 and 2*N*1*Q*.

Similarity clustering. Structure is generally more conserved than sequence, hence structural similarity should ideally be used to split the data set into training, test and validation data sets that share as little homology as possible. However, similarity scores based on RMSE have the problem that global rearrangements dominate this score even if all structural domains are conserved perfectly. We therefore decided to use sequence similarity as a proxy for structural similarity and hence functional similarity. We aligned all sequences of all chains in the data set with a scoring function that rewarded matches with 1 and punished mismatches and opening gaps with -1. Gap extensions were treated as neutral. Then, we summed up the scores for structures that had multiple chains, and divided the scores by the overall length of the longest of each compared sequence. Using these length-specific matching scores, we used complete hierarchical clustering and chose a reasonable cutoff (0.7) to obtain sequence clusters.

Data sampling. First, we sampled uniformly over all sequence clusters. Longer sequences are more informative than shorter sequences, hence we sampled proportional to sequence length.

Homologs. To obtain sequences homologous to those in our PDB structure data set, we followed the following workflow. First, we downloaded homologous sequences from RNAcentral [54] using a Python script published on the RNAcentral Sequence search API website at https://rnacentral.org/sequence-search/api. Then, we only kept homologous sequences that fulfilled the following conditions (i) their E-value had to be less than 0.01, (ii) they must not have insertions, and (iii) they had to have at least one mutation other than a deletion. Gaps in homologous sequences were filled up with the letter 'N'.

Deep learning architectures and hyperparameters

VQ-VAE network architecture. We encoded RNA tertiary structures by distance tensors ($L \times L \times 64$)—as explained above—that act as both input and target of our VQ-VAE. We zero-padded all entries of distance tensors that were outside the maximum sequence length $L = 100$. Following the original work of the VQ-VAE [33], the encoder architecture was a residual network [55] that consisted of one convolutional layer followed by 4 residual blocks. Each residual block was of the form: Block = 2x[Batchnorm, ReLU Activation, Conv]. The two convolutional layers ("Conv") within each block could have different convolutional kernels, whose sizes we report in the following format hereafter: [hight × width, out_channels], with block and kernel settings as in Tables 2 and 3

In the vector quantisation step, the encoder's residual network output of shape ($L \times L \times 8$) was mapped to VQ-VAE encodings (distance classes) of shape ($L \times L \times 3$) using 3 codebook vectors of embedding dimension 8. We enforced symmetry of the distance class tensors (along the first 2 dimensions) by adding the transpose of the embedding and dividing the result by 2.

Table 2. VQ-VAE encoder architecture.

Layer type	Kernel 1	Kernel 2
Conv. Layer	[7 × 7, 32]	
Res. Block	[5 × 5, 16]	[3 × 3, 8]
4 × Res. Block	[5 × 5, 8]	[3 × 3, 8]

https://doi.org/10.1371/journal.pone.0297105.t002

The decoder architecture took the VQ-VAE encodings (distance class tensor) as input and consisted of four sets of residual blocks followed by a final ReLU layer.

Between each set of residual blocks, we up-scaled the number of feature-maps by inserting an additional convolutional layer that doubled the number of feature maps. The output of the decoder had shape ($L \times L \times 64$) and corresponded to the distance tensor reconstruction. For training, we used the same learning objective as in the original VQ-VAE [33] using an exponential moving average for the codebook vector updates with a decay rate of 0.99. For optimization, we used standard Adam Optimizer [56] with a learning rate of 1×10^{-5} at a batch size of 100.

Generator network: Data preprocessing. We stacked the following 4 tensors to obtain input tensors for the generator network: (i) an encoded RNA sequence tensor, (ii) the corresponding, partially masked distance class tensor, (iii) coordinate frame tensors to provide the network with positional information, and (iv) an optional attention map of homologous sequence alignment and SHAPE data. RNA sequences of length L were encoded as unique bit patterns of shape ($L \times L \times 8$), (see S1 Fig). First, an RNA sequence was one-hot encoded by a tensor of shape ($L \times 4$). Unknown nucleotides that were denoted by an "N" in the sequence were encoded by setting all values in the one-hot encoding to 0.25. This one-hot encoded sequence was then copied $L - 1$ times to obtain a tensor of shape ($L \times L \times 4$). Then, this tensor and its transpose were stacked along the last dimension to obtain a tensor of shape ($L \times L \times 8$). This tensor corresponded to a unique bit pattern for each possible pairing and also contained directional information. For sequences with $L < 100$, the sequence tensors were uniformly padded with −1.

Partially masked distance class tensors were obtained from the distance classes using a pretrained VQ-VAE, followed by a "partial masking" process in which some pixels were set to zero as described in the following. First, we drew the number of pixels to be masked at a given training step from a truncated normal distribution with mean $L^2/2$, standard deviation $L^2/4$ and which was bounded at 2 standard deviations around the mean (this ensured that at least 0 and at most L^2 pixels were masked, while rarely masking either very few or very many pixels).

Table 3. VQ-VAE decoder architecture.

Layer type	Kernel 1	Kernel 2
Conv. Layer	[5 × 5, 8]	
4×Res. Block	[5 × 5, 8]	[5 × 5, 8]
Conv. Layer	[5 × 5, 16]	
4 × Res. Block	[5 × 5, 16]	[5 × 5, 16]
Conv. Layer	[5 × 5, 32]	
4 × Res. Block	[5 × 5, 32]	[5 × 5, 32]
Conv. Layer	[5 × 5, 64]	
4 × Res. Block	[3 × 3, 64]	[3 × 3, 64]
ReLU Activation		

https://doi.org/10.1371/journal.pone.0297105.t003

Then, we randomly drew the corresponding number of pixel positions and encoded masked pixels with (0, 0, 0). However, we found that encoding masked pixels this way made it difficult for the network to learn the difference between masked distance classes (0, 0, 0) and the regular, non-masked distance classes (1, 0, 0), (0, 1, 0) or (0, 0, 1). To overcome this limitation, we encoded all regular, non-masked distance classes by setting all zeros to −1, so that, for example, (1, 0, 0) was encoded as (1, −1, −1). Coordinate frames consisted of a "diagonal frame" that had ones along the diagonal and was zero elsewhere, and a "padding frame" which contained a box of side length L that had ones along its border and was zero elsewhere. The coordinate frames were padded with −1 when sequences were shorter than 100 nt. The production of the attention map of homologous sequences and SHAPE data is described in the following section in detail. Overall, the generator input was a tensor of shape ($L \times L \times 14$), and contained the encoded sequence tensor of shape ($L \times L \times 8$), the partially masked distance tensor of shape ($L \times L \times 3$), both coordinate frames of shape ($L \times L \times 2$) and the homologous sequences alignment and SHAPE data attention map tensor of shape ($L \times L \times 1$).

Generator network: Attention map of homologous sequences and SHAPE. For each training example, an array of 50 randomly chosen, aligned and one-hot encoded homologous sequences was produced. While the 4 standard nucleotides were encoded using one-hot coding (e.g. "A" was encoded as (1, 0, 0, 0)), gaps and unknown nucleotides were encoded by setting all possible one-hot coding values to 0.25 such that the one-hot coding vector became (0.25, 0.25, 0.25, 0.25). When there were fewer than 50 homologous sequences, the original sequence was used to "fill up" the array. The resulting array of one-hot encoded homolgeous sequences had shape ($L \times 50 \times 4$). Using a standard dense layer, this array was mapped onto a tensor of shape ($L \times 50$). Then, simulated SHAPE data was included by stacking a vector containing L SHAPE reactivity values on that array, resulting in a tensor of shape ($L \times 51$). Using two separate dense layers, two tensors of shape ($L \times 64$) were produced and self-attention was applied to these tensors by using them as query and key tensors as described in the original Transformer paper [42], resulting in an attention map of shape ($L \times L \times 1$). Self-Attention added the benefit, that for every pixel, the corresponding nucleotide could receive both homologous sequence information as well as SHAPE reactivity information from all other nucleotides in that sequence. We computed results on a sample attention map for the Adenine Riboswitch (PDB: 1y26) in S2 Fig.

Generator network: Architecture. Using the ($L \times L \times 14$) input tensor described above, the generator network performed regression on the full distance class map under the masked learning objective, which is specified further below. The input tensor was first passed through a convolutional layer to be further processed by a deep residual network architecture with eight blocks, each one having the following structure: ResBlock = 2x[Batchnorm, Elu Activation, Conv] + 1x[Batchnorm, Elu Activation, Conv] + 1x[Batchnorm, Elu Activation, DilatedConv], with standard skip connections after the first two and between the last two subblocks. We employed standard convolutions with kernel sizes as described in Table 4 below. We also

Table 4. Generator network: Resiudal architecture.

Layer type	Conv	DilatedConv
Conv. Layer	[3 × 3, 26]	
8 × ResBlock	[3 × 3, 26]	[3 × 3, 26], dilation rate = 2
Batch normalisation		
Elu Activation		
Conv. Layer	[1 × 1, 3]	
Softmax Activation		

https://doi.org/10.1371/journal.pone.0297105.t004

Table 5. Generator network: Optimization hyperparameters.

LazyAdamOptimizer	$\beta_1 = 0.9, \beta_2 = 0.997, \epsilon = 1e-8$
Batch size	500
Weight decay	0.01
Learning rate α	0.001
Learning rate warmup steps	100000

https://doi.org/10.1371/journal.pone.0297105.t005

used dilated convolutions with a dilation rate of 2 in the residual blocks. Here, the last convolution layer compressed the residual network output into an appropriate shape of $(L \times L \times 3)$ so that, after a final softmax activation, the network's output corresponded to predicted distance classes, see Table 4.

The network was trained under the masked learning objective, implemented using the cross entropy between target distance classes and generator predictions coming from input tensors with partially masked distance tensors as a loss function. We further employed weight decay (L_2 regularization) following standard recommendations for training large residual architectures [55]. Table 5 shows the generator hyperparameter setup using a lazy Adam Otimizer.

For stabilized training, we chose a learning rate with linear warmup scaling and cosine decay. After five million iteration steps we stopped the network training.

Score model. The Score Model was implemented to distinguish which one of two distance class maps was better. Its architecture as described in Table 6 was a residual network using optimization hyperparameters as further described in Table 7.

The Score Model was trained to discriminate between "correct" and "incorrect" distance class maps as described in the following. The set of correct examples consisted of all 8048 examples in the training data set. To create the set of incorrect examples, we used SimRNA-based data augmentation to compute 100 drift structures of up to 10 Å RMSE away from each original structure in the training data set. Using the VQ-VAE setup, we then computed

Table 6. Score model network architecture.

Layer type	Conv	DilatedConv
Conv. Layer	[3 × 3, 5]	
4 × ResBlock	[3 × 3, 5]	[3 × 3, 5], dilation rate = 2
Batch normalisation		
Relu Activation		
Conv. Layer	[1 × 1, 100]	
Relu Activation		
Conv. Layer	[1 × 1, 1]	

https://doi.org/10.1371/journal.pone.0297105.t006

Table 7. Score model network: Optimization hyperparameters.

LazyAdamOptimizer	$\beta_1 = 0.9, \beta_2 = 0.997, \epsilon = 1e-9$
Batch size	100
Weight decay	0.0001
Learning rate α	0.001
Learning rate decay	0.9988

https://doi.org/10.1371/journal.pone.0297105.t007

distance class maps for all original as well as drift structures. We further increased the dataset of incorrect examples by factor 10 by randomly flipping class pixels from the targets distance classes. We also used the Generator networks' argmax predictions, showing only 0%, 5%, 10%, 15% and 20% of the target distance classes at the input. Then, at each training step, both a correct and a corresponding incorrect distance class map of shape ($L \times L \times 3$) were fed through the network to obtain two "logit maps" of shape ($L \times L \times 1$). The values of the correct logit map were then subtracted from corresponding values of the incorrect logit map, followed by taking the sum over all differences. We optimized this objective with L_2 regularization under the following hyperparameter setup and used a standard learning rate decay exponential to the iteration steps.

MCTS: Sampling structural ensembles. The MCTS algorithm sampled pixelwise one of the three distance classes using the Generator network iteratively. Naturally, the diagonal had high class probabilitiy values, so that we started initialising the search tree by setting the diagonal to the nearest distance class. Except for the diagonal, we started by masking all ($L \times L \times 3$) target distance classes with zeros and iteratively filling in class indicators, e.g. (1, −1, −1) if the MCTS chose the first distance class. With that basic step logic, MCTS could iteratively fill up pixels and update visit counts and values at each node. The value objective in the MCTS was designed such that the network aimed to get a sharp view after some pixels were set, so that not every single of the $L^2/2$ pixels needed to be sampled. This reduced the depth of the search tree by a large margin. Typically, the Generator network produced sufficiently sharp predictions when the MCTS was able to predict 30% of the target distance classes correctly. The remaining pixels were then filled up using an argmax prediction of the Generator network given that leaf. Exemplarly, we computed the leafs of the search tree for two alternative structures shown in S3 Fig for the ZMP- Riboswitch (PDB: 4xw7).

Refinement. The SimRNA simulation runs are carried out under the standard configuration file:

```
#  config.dat
NUMBER_OF_ITERATIONS  2000000
TRA_WRITE_IN_EVERY_N_ITERATIONS  200000

INIT_TEMP  1.15
FINAL_TEMP  0.9

BONDS_WEIGHT  1.0
ANGLES_WEIGHT  1.0
TORS_ANGLES_WEIGHT  0.0
ETA_THETA_WEIGHT  0.40
```

The computation of the five spectral clusters required the following steps:

```
Example:  7elq:  GGAGUAGAAGCGUUCAGCGGCCGAAAGGCCGCCCGGAAAUUGCUCC
INPUT:  Sequence
1.  MCTS:  sample  structural  ensemble  using  generative  network
    Output:  N  structures
    python  mcts.py  GGAGUAGAAGCGUUCAGCGGCCGAAAGGCCGCCCGGAAAUUGCUCC
2.  SimRNA:
for  i  in  1..N:
    ./SimRNA  7elq_i.pdb  -c  config.dat
    #  compute  1000  repition  runs
```

```
    #  ->  N*1000  trafl  files  /  pdb  outputs
    #  ->  concatenate  all  trafl  files
Output:  7elq.trafl
3.  SimRNA  Clustering
    ./clustering  7elq.trafl  0.25  5.0  10.0  15.0  20.0  25.0
    Output:  5  cluster  centers
4.  Discriminator:
    Rank  5  cluster  centers
OUTPUT:  5  ranked  pdb  files  (cluster  centers)
```

Supporting information

S1 Fig. Encoding of sequence information. RNA sequences of length L were encoded as unique bit patterns of shape ($L \times L \times 8$): **A** First, every nucleotide in an RNA sequence was one-hot encoded, e.g. *G*: (1, 0, 0, 0), *C*: (0, 1, 0, 0), *A*: (0, 0, 1, 0), *U*: (0, 0, 0, 1), *N*: (0.25, 0.25, 0.25, 0.25). For the full sequence, these one-hot encodings led to a tensor of shape ($L \times 4$). Unknown nucleotides that were denoted by an "N" in the sequence were encoded by setting all values in the one-hot encoding to 0.25. This one-hot encoded sequence was then copied L times (a1) to obtain a tensor of shape ($L \times L \times 4$). Then, this tensor and its transpose (a2) were stacked along the last dimension to obtain a tensor of shape ($L \times L \times 8$). **b** A sample sequence Tensor that corresponded to a unique bit pattern for each possible pairing and also contained directional information. For sequences with $L < 100$, the sequence tensor was uniformly padded with −1 Red insert: example bit pattern for the Tensor at the first three pixels with depth 8.
(PDF)

S2 Fig. Attention: Improved performance from adding structural probing data (SHAPE) and homologous sequences. Attention Maps were a good indicator for the location of structural contacts. High attention values (red pixels) were almost exclusively found at the "near" distance class, resulting in higher probability scores for the generator's initial prediction for that class. Incorporation of experimental **A** or simulated **B** SHAPE data both resulted in lower false positive rates of attention placement compared to when no SHAPE data **C** or no SHAPE data and no homolgous sequence information **d** was used, with false positive rates being, 4.3%, 5.7%, 14.8% and 21.4%, respectively (the false positive rate was calculated as the number of incorrectly placed red, high attention points divided by the total number of red, high attention points, where high attention points were defined by attention scores above 0.01).
(PDF)

S3 Fig. MCTS: Search tree with two distinct leafs. Starting with all target distance classes masked, the Generator places initial probabilities for every pixel in the distance class softmax prediction. From there, pixels were sampled iteartively using the MCTS search objective which aimed for entropy reduction. We derived two terminal leaf nodes **A** for which the Generator network saw enough distance pixels to fill up the remainers using its argmax prediciton **B**. From those filled up leafs, the VQ-VAE could be applied to decode into real distance space **C**. After further energy refinement, we showed for the ZMP-Riboswitch (PDB: 4XW7), that those two leafs indeed corresponded to two different structures. The riboswitch has a movable hinge part, which gets stabilized by a small molecule. Hence, our best leaf prediction in red is closer to the blue target solution, bottom **C**. We also sampled an alternative structure. In this

particular example, the strechted grey RNA structure **C** was ranked with a lower score by the Score Model.
(PDF)

Acknowledgments

Computational support and infrastructure was provided by the Center for Information and Media Technology (ZIM) at the Heinrich-Heine-Universität Düsseldorf, Germany. Therefore, we would like to thank the high performance computing (HPC) team and infrastructure at Heinrich-Heine-Universität Düsseldorf for assisting in the computational experiments.

Author Contributions

Conceptualization: Julius Ramakers, Christopher Frederik Blum, Sabrina König, Stefan Harmeling, Markus Kollmann.

Data curation: Julius Ramakers, Christopher Frederik Blum.

Formal analysis: Julius Ramakers, Markus Kollmann.

Funding acquisition: Christopher Frederik Blum, Stefan Harmeling, Markus Kollmann.

Investigation: Christopher Frederik Blum, Sabrina König, Markus Kollmann.

Methodology: Christopher Frederik Blum, Markus Kollmann.

Project administration: Christopher Frederik Blum, Stefan Harmeling, Markus Kollmann.

Resources: Stefan Harmeling, Markus Kollmann.

Software: Julius Ramakers, Sabrina König.

Supervision: Markus Kollmann.

Validation: Julius Ramakers.

Visualization: Julius Ramakers, Sabrina König.

Writing – original draft: Julius Ramakers, Christopher Frederik Blum, Markus Kollmann.

Writing – review & editing: Julius Ramakers, Christopher Frederik Blum, Stefan Harmeling, Markus Kollmann.

References

1. Zhang, Qi D, Al-Hashimi HM et al. Visualizing spatially correlated dynamics that directs RNA conformational transitions. Nature 450.7173 (2007): 1263–1267. https://doi.org/10.1038/nature06389 PMID: 18097416
2. Dethoff Elizabeth A., Chugh Jeetender, Mustoe Anthony M. and Al-Hashimi Hashim M. Functional complexity and regulation through RNA dynamics. Nature 482, 322–330 (2012). https://doi.org/10.1038/nature10885 PMID: 22337051
3. Kulkarni J.A., Witzigmann D., Thomson S.B., Chen S., Leavitt B.R., Cullis P.R., et al. The current landscape of nucleic acid therapeutics. Nat Nanotechnol. 2021; 16:630–643. https://doi.org/10.1038/s41565-021-00898-0 PMID: 34059811
4. Damase TR, Sukhovershin R, Boada C, Taraballi F, Pettigrew RI, Cooke JP. The Limitless Future of RNA Therapeutics. Frontiers in Bioengineering and Biotechnology. 2021; 9:161. https://doi.org/10.3389/fbioe.2021.628137 PMID: 33816449
5. Nance KD, Meier JL. Modifications in an Emergency: The Role of N1-Methylpseudouridine in COVID-19 Vaccines. ACS Central Science. 2021; 7(5):748–756. https://doi.org/10.1021/acscentsci.1c00197 PMID: 34075344

6. Li B, Cao Y, Westhof E, Miao Z. Advances in RNA 3D Structure Modeling Using Experimental Data. Frontiers in Genetics. 2020; 11:1147. https://doi.org/10.3389/fgene.2020.574485 PMID: 33193680
7. Watkins AM, Rangan R, Das R. FARFAR2: Improved De Novo Rosetta Prediction of Complex Global RNA Folds. Structure. 2020; 28(8):963–976.e6. https://doi.org/10.1016/j.str.2020.05.011 PMID: 32531203
8. Antczak M., Adamiak R.W., Szachniuk M, et al. New functionality of RNAComposer: an application to shape the axis of miR160 precursor structure. Acta Biochimica Polonica. 2016: 63(4):737–744. https://doi.org/10.18388/abp.2016 1329 PMID: 27741327
9. Zhao Y. et al. Automated and fast building of three-dimensional RNA structures. Sci Rep. 2012; 2:734. Epub 2012 Oct 15. https://doi.org/10.1038/srep00734 PMID: 23071898
10. Yi Zhang, Jun Wang, Yi Xiao 3dRNA: 3D structure prediction from linear to circular RNAs. Journal of Molecular Biology. 2022. https://doi.org/10.1016/j.jmb.2022.167452
11. Alford RF, Leaver-Fay A, et al. The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. Journal of Chemical Theory and Computation. 2017; 13(6):3031–3048. https://doi.org/10.1021/acs.jctc.7b00125 PMID: 28430426
12. Xiong P., Wu R., Zhan J. and Zhou Y. Pairing a high-resolution sta- tistical potential with a nucleobase-centric sampling algorithm for improving RNA model refinement. Nat Commun 12, 2777 (2021). https://doi.org/10.1038/s41467-021-23100-4
13. Boniecki MJ, Lach G, Dawson WK, Tomala K, Lukasz P, Soltysinski T, et al. SimRNA: a coarse-grained method for RNA folding simulations and 3D structure prediction. Nucleic Acids Res. 2016; 20; 44(7): e63. https://doi.org/10.1093/nar/gkv1479 PMID: 26687716
14. Hazapi, O. et al. Advances in RNA 3D Structure Prediction. Handbook of Machine Learning Applications for Genomics. Studies in Big Data, vol 103. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-9158-4_12
15. Xiujuan Ou, Yi Zhang, Yiduo Xiong, and Yi Xiao. Machine Learning: A Tool to Shape the Future of Medicine. Journal of Chemical Information and Modeling 62.23 (2022): 5862–5874. https://doi.org/10.1021/acs.jcim.2c00939
16. Townshend, Raphael J. L. and Eismann, Stephan and Watkins, Andrew M. and Rangan, Ramya and Karelina, Maria and Das, Rhiju , et al. Geometric deep learning of RNA structure Science. 2021. https://doi.org/10.1126/science.abe5650
17. Jiayang Chen and Zhihang Hu and Siqi Sun and Qingxiong Tan and Yixuan Wang and Qinze Yu, et al. Interpretable RNA Foundation Model from Unannotated Data for Highly Accurate RNA Structure and Function Prediction. arxiv preprint: https://arxiv.org/abs/2204.00300
18. Laiyi Fu, Yingxin Cao, Jie Wu, Qinke Peng, Qing Nie, Xiaohui Xie UFold: fast and accurate RNA secondary structure prediction with deep learning. Nucleic Acids Research, Volume 50, Issue 3, 22 February 2022, Page e14. https://doi.org/10.1093/nar/gkab1074
19. Jaswinder Singh, Kuldip Paliwal, Thomas Litfin, Jaspreet Singh, Yaoqi Zhou Predicting RNA distance-based contact maps by integrated deep learning on physics-inferred secondary structure and evolutionary-derived mutational coupling. Bioinformatics, Volume 38, Issue 16, August 2022, Pages 3900–3910. https://doi.org/10.1093/bioinformatics/btac421
20. Li Y., Zhang C., Feng C. et al. Integrating end-to-end learning with deep geometrical potentials for ab initio RNA structure prediction. Nat Commun 14, 5745 (2023). https://doi.org/10.1038/s41467-023-41303-9 PMID: 37717036
21. Li B, Cao Y, Westhof E, Miao Z., Magnus M., et al. Advances in RNA 3D Structure Modeling Using Experimental Data. RNA. 2012: 2012 Apr; 18(4):610–25. https://doi.org/10.1261/rna.031054.111
22. Li B, Cao Y, Westhof E, Miao Z., Magnus M., et al. Advances in RNA 3 Structure D Modeling Using Experimental Data. RNA. 2015 Jun; 21(6):1066–84. https://doi.org/10.1261/rna.049502.114
23. Li B, Cao Y, Westhof E, Miao Z., Magnus M., et al. Advances in RNA 3D Structure Modeling Using Experimental Data. RNA. 2017:May; 23(5):655–672. https://doi.org/10.1261/rna.060368.116
24. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021. https://doi.org/10.1038/s41586-021-03819-2
25. Baek M, DiMaio F, Anishchenko I, Dauparas J, Ovchinnikov S, Lee GR, et al. Accurate prediction of protein structures and interactions using a three-track neural network. Science. 2021. https://doi.org/10.1126/science.abj8754 PMID: 34282049
26. Morcos F, Pagnani A, Lunt B, Bertolino A, Marks DS, Sander C, et al. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. Proceedings of the National Academy of Sciences. 2011; 108(49):E1293–E1301. https://doi.org/10.1073/pnas.1111471108 PMID: 22106262
27. Marks D, Hopf T, Sander C. Protein structure prediction from sequence variation. Nat Biotechnol. 2012; 30:1072–1080. https://doi.org/10.1038/nbt.2419 PMID: 23138306

28. Schneideret al. When will RNA get its AlphaFold moment? Nucleic Acids Res. 2023 Oct 13; 51 (18):9522–9532. https://doi.org/10.1093/nar/gkad726 PMID: 37702120; PMCID: PMC10570031

29. Senior AW, Evans R, Jumper J. Improved protein structure prediction using potentials from deep learning. Nature. 2020; 577:706–710. https://doi.org/10.1038/s41586-019-1923-7 PMID: 31942072

30. Miao Z, Westhof E. RNA Structure: Advances and Assessment of 3D Structure Prediction. Annual Review of Biophysics. 2017; 46(1):483–503. https://doi.org/10.1146/annurev-biophys-070816-034125 PMID: 28375730

31. Mortimer SA, Weeks KM. A Fast-Acting Reagent for Accurate Analysis of RNA Secondary and Tertiary Structure by SHAPE Chemistry". J Am Chem Soc. 2007; 129(14):4144–45. https://doi.org/10.1021/ja0704028 PMID: 17367143

32. Tijerina P, Mohr S, Russell R. DMS footprinting of structured RNAs and RNA-protein complexes". Nat Protoc. 2007; 2(10):2608–23. https://doi.org/10.1038/nprot.2007.380 PMID: 17948004

33. van den Oord A, Vinyals O, Kavukcuoglu K. Neural Discrete Representation Learning. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17. Red Hook, NY, USA: Curran Associates Inc.; 2017. p. 6309–6318.

34. Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. CoRR. 2018;abs/1810.04805.

35. Yang Z, Dai Z, Yang Y, Carbonell JG, Salakhutdinov R, Le QV. XLNet: Generalized Autoregressive Pretraining for Language Understanding. Advances in neural information processing systems 32 (2019), p. 5754–5764.

36. Oord Avd, Kalchbrenner N, Vinyals O, Espeholt L, Graves A, Kavukcuoglu K. Conditional Image Generation with PixelCNN Decoders. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. NIPS'16. Red Hook, NY, USA: Curran Associates Inc.; 2016. p. 4797–4805.

37. He, Kaiming, et al. Masked autoencoders are scalable vision learners. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

38. Nalisnick E, Matsukawa A, Teh YW, Gorur D, Lakshminarayanan B. Do Deep Generative Models Know What They Don't Know? In: International Conference on Learning Representations; 2019.

39. Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT press; 2018.

40. Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press; 2016.

41. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015; 521(7553):436–444. https://doi.org/10.1038/nature14539 PMID: 26017442

42. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is All you Need. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, et al., editors. Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc.; 2017.

43. Kaplan J, McCandlish S, Henighan T, Brown TB, Chess B, Child R, et al.. Scaling Laws for Neural Language Models; 2020.

44. Adamczyk B., Antczak M., Szachniuk M.. RNAsolo: a repository of clean, experimentally determined RNA 3D structures. Bioinformatics 38(14):3668–3670. https://doi.org/10.1093/bioinformatics/btac386

45. Leontis, Westhof Nonredundant 3D Structure Datasets for RNA Knowledge Extraction and Benchmarking. RNA 3D Structure Analysis and Prediction. (Vol. 27, pp. 281–298). Springer Berlin Heidelberg https://doi.org/10.1007/978-3-642-25740-7-13

46. Pereyra, Gabriel, et al. Regularizing neural networks by penalizing confident output distributions. arXiv preprint arXiv:1701.06548 (2017)

47. Müller Rafael, Kornblith Simon, and Hinton Geoffrey E. When does label smoothing help? Advances in neural information processing systems 32 (2019).

48. Tero Karras, et al. Elucidating the Design Space of Diffusion-Based Generative Models Proc. NeurIPS. 2022

49. Azadi S, Olsson C, Darrell T, Goodfellow I, Odena A. Discriminator Rejection Sampling; 2019.

50. Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, et al. Mastering the game of Go without human knowledge. Nature. 2017; 550(7676):354–359. https://doi.org/10.1038/nature24270 PMID: 29052630

51. Miao Z, Adamiak RW, Antczak M, Boniecki M, Bujnicki J, Chen SJ, et al. RNA-Puzzles Round IV: 3D structure predictions of four ribozymes and two aptamers. RNA (New York, NY). 2020; 26(8):982–995. https://doi.org/10.1261/rna.075341.120 PMID: 32371455

52. Hajdin CE, Bellaousov S, Huggins W, Leonard CW, Mathews DH, Weeks KM. Accurate SHAPE-directed RNA secondary structure modeling, including pseudoknots. Proceedings of the National Academy of Sciences. 2013; 110(14):5498–5503. https://doi.org/10.1073/pnas.1219988110 PMID: 23503844

53. Carrascoza F, Antczak M, Miao Z, Westhof E, Szachniuk M. Evaluation of the stereochemical quality of predicted RNA 3D models in the RNA-Puzzles submissions. RNA. 2022 Feb; 28(2):250–262. https://doi.org/10.1261/rna.078685.121 PMID: 34819324; PMCID: PMC8906551

54. Petrov AI, Kay SJE, Gibson R, Kulesha E, Staines D, Bruford EA, et al. RNAcentral: an international database of ncRNA sequences. Nucleic Acids Res. 2015; 43(Database issue):D123–129. https://doi.org/10.1093/nar/gku991 PMID: 25352543

55. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016; p. 770–778.

56. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. CoRR. 2015;abs/1412.6980.