AutoRNA: RNA tertiary structure prediction using variational autoencoder.

M.A. Kazanskii[1], L. Uroshlev[1], F. Zatylkin[1], I. Pospelova[1], O.Kantidze[1], Y.Gankin[1]

[1] Quantori LLC, Cambridge, MA, USA

*Corresponding author: mkazanskii@gmail.com, maxim.kazanskiy@quantori.com

Abstract

Understanding the RNA tertiary structure is crucial for advancing therapeutic development and vaccine design. Previous methods, such as dynamic simulations, have been used to study RNA structure. However, these methods often struggle to capture the complex non-linear relationships in RNA sequences. Other approaches focused on simple models, resulting in lower performance. Indeed, one of the main challenges in predicting RNA tertiary structure is the lack of data, and using complicated algorithms is limited. In this paper, we introduce the Variational Autoencoder (VAE). In this paper, we introduce the Variational Autoencoder (VAE) for the problem of RNA folding and demonstrate that the Variational Autoencoder could yield promising results. Our model achieves a Root Mean Square Error (RMSE) of approximately 3.3Å in predicting nucleotide positions, significantly outperforming existing approaches.

Introduction

For developing new RNA-based drugs and vaccines, it is crucial to understand the fundamental properties of various RNA molecules [1], including their three-dimensional (3d) structure. Leveraging machine learning (ML) techniques to decipher protein folding can pave the way for resolving comparable challenges in the RNA domain. AlphaFold2 [2] showed the power of machine learning methods for solving protein folding problems. Nevertheless, adapting neural networks similar to AlphaFold2 for RNA could present challenges. On the one side, the RNA folding problem might seem less complex due to a smaller variety of nucleotides compared to the amino acids in proteins, which reduces the number of viable options. However, the datasets of RNA structures available for training are significantly smaller than those for proteins [3], which may hinder the effectiveness of alignment-based ML approaches [2] when applied to RNA. As of this writing, about 4500 structures in the PDB contain RNA and over 200000 protein structures. Therefore, the applicability of machine learning methods based on the coevolution of monomers, like AlphaFold2[2], for RNA cases may be limited. Studies have also indicated that the energy landscape of 3D RNA structures could be more intricate than that of proteins [3], with RNA potentially exhibiting a more substantial number of local energy minima. Furthermore, understanding of nucleotide coevolution in RNA is limited, mainly because RNA encompasses distinct classes such as tRNA, miRNA, and siRNA, each with its own evolutionary path. Moreover, the way in which RNA evolved for procaryotes, eukaryotes, and viruses were quite different due to the different mechanisms of regulation and translation.

As a result, for predicting of 3D RNA structures, it is necessary to employ ML techniques that do not solely rely on alignment or empirical knowledge of nucleotide coevolution. Precise determination of 3D structure requires ML methods that either do not use alignment and coevolutionary data or utilize such information with a high degree of caution.

Here, we present a novel simple architecture for predicting the 3D structure of RNA sequences. Our approach is based on the variational autoencoder. Since the dataset is small, a simple generative model is justified. We were able to reach an average RMSE of approximately 3.3Å.

Related Work

Numerous algorithms have been developed for the 3D structure prediction of RNA, though the accuracy of these predictions is not yet exceptionally high. Starting from the problem of predicting the secondary structure of RNA, several attempts have been made to generalize methods and algorithms for the 3D case. In [4], the authors tried to apply methods of dynamic programming for 3D structure prediction The most utilized techniques are based on the empirical potential method and its variants, which calculate the interaction forces between RNA atoms by examining interatomic contacts [6,7]. In the methods, the interaction forces between RNA atoms are determined from the analysis of interatomic contacts. This approach is also used for protein or protein-ligand interaction modeling [8]. Notable empirical potentials include AMBER [9] and CHARMM [10], which are integrated into widely used molecular dynamics software. However, employing molecular dynamics directly for modeling is a time-consuming process. The complete folding of an RNA molecule requires 10^{-6} seconds for forming the 3D backbone structure and 10^{-4} seconds for folding each nucleotide in the molecule [11]. Therefore, the folding time could be 0.01-0.1 second for a molecule of several tens of nucleotides, so direct folding of such structures could be expensive. Coarse-grained potentials are one of the possible ways [12] to reduce complexity and speed up computations.

In recent times, hybrid methods have become more popular. These methods combine empirical potentials with various neural network structures. For instance, one study has applied the transformer architecture in conjunction with Rosetta's empirical potentials [12], while another has merged coarse-grained empirical potential with Generative Adversarial Networks (GANs) [13].

An alternative approach derived from protein design is the assembly of structures using elementary templates. A technique described in one of the studies involves breaking down structures into a collection of 3D templates and then optimizing these through a minimization process in the final phase of the algorithm. Reference [14] describes a web server that decomposes structures into a set of 3D templates followed by a minimization process in the final stage of the algorithm.

In addition, various kinds of machine learning are used to predict the structure. In [15], a modified U-net neural network predicts a distance map for 3D RNA structure. A promising approach was shown in [18] where structure sampling of FARFAR2 and Rosetta is combined with deep learning score.

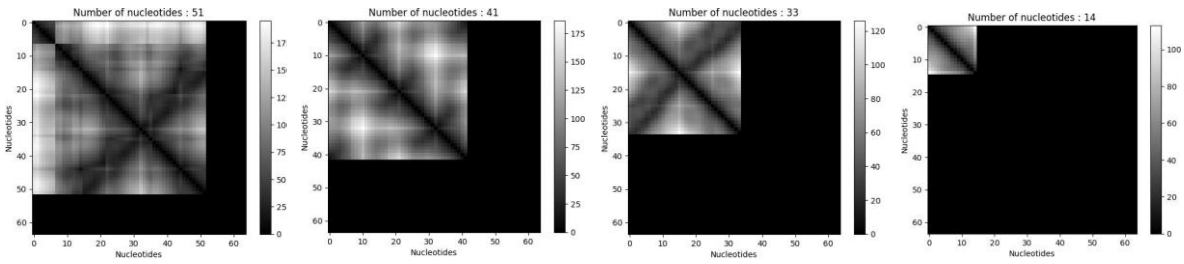
Methodology

Data preprocessing

We used Protein Data Bank (PDB) dataset, which contain 3D structural information of the RNA molecules [3]. The data processing involved several steps: we removed the duplicates and the empty sequences, dropped molecules from the chain, and calculated the spatial positions of the backbone of each of the nucleotides. We used the backbone coordinates as a tertiary structure for the RNA sequence. For the experiment, we selected only those RNA sequences that were 64 nucleotides or shorter. The distance matrix was generated to represent the distances between each and other nucleotide backbone. For the nucleotide sequences with a number of nucleotides (or sequence length) less than 64, we used padding with a value of 0. Additionally, we clipped the values of the distance. Indeed, the distance mostly lay in the interval (0 Å,80 Å), and only 0.06% of the distances exceeded the value of 100Å. Therefore, fixing the values to 100 for the distances exceeding 100 would not significantly affect the algorithm's performance. The examples of the distance matrices are shown in Figure 1. The distance matrices were normalized in the range of (0,1) for the input to the autoencoder. The values of the normalized distance matrixes were determined as:

$$dm_{norm} = \frac{dm}{100}$$

Figure 1. Examples of the input images to the ML pipeline. Each image is equivalent to the distance matrix corresponding to the RNA sequence with zero padding. The images were scaled by a factor of 255 for better visualization: original images have pixel values in range (0,1). On the X and Y axes, the nucleotides are represented. For the sake of sanity, we have not included the sequences. The black area corresponds to the padding (to reach 64 nucleotides).



Together with the distance matrices of the RNA sequences, we use nucleotide letters ("A", "G", "T", "C") as one-hot encoder vectors with the following rules: A -> (1, 0, 0, 0), G -> (0, 1, 0, 0), T -> (0, 0, 1, 0), C -> (0, 0, 0, 1). We added padding with values (0, 0, 0, 0) to the sequences of the length less than 64. The dataset was divided into [train/validation/test] subsets with a proportion [0.80 / 0.15 / 0.05]. The total number of sequences was [2151 / 404 / 134] for [train/validation/test] subsets.

Model

We used the variational autoencoder [17] to predict the distance matrix. In particular, we have the conditions vector concatenated with the random vector as input. The condition vector is the flattened one-hot encoded nucleotide sequence (size of 256 nodes). The random greyscale image is a noise vector

of the same size as the distance matrix (64*64 = 4096 nodes). We used a traditional variational autoencoder that consists of an encoder, bottleneck, and decoder. The encoder consists of two fully connected layers with a number of nodes equal to 4096 and 2048. The Bottleneck layer is trivial and consists of only one layer of size 1024. The decoder consists of three fully connected layers of size 2048, 4096, and 4096. The last layer represents the values of the distance matrix. After each fully connected layer (besides the last one), we used batch normalization, which improved the performance and added stability. In addition, we applied dropout with a rate of 0.5 for the output signals from the batch-normalization layers. The total number of trainable parameters in the model equals to 58 238 976.

The loss function for the model is chosen to be classical for autoencoders and consists of two parts: the reconstruction loss and the Kullback–Leibler divergence loss [19]. We used Mean Average Error as a reconstruction loss due to its stability to outliers (in comparison with Mean Square Error Loss) [20]. We used the scaling parameter β [21] between two components as one of the hyper-parameters for the model. The loss function is presented below:

$$L_{\beta-VAE} = L_{RECON} + \beta \cdot L_{KL}$$

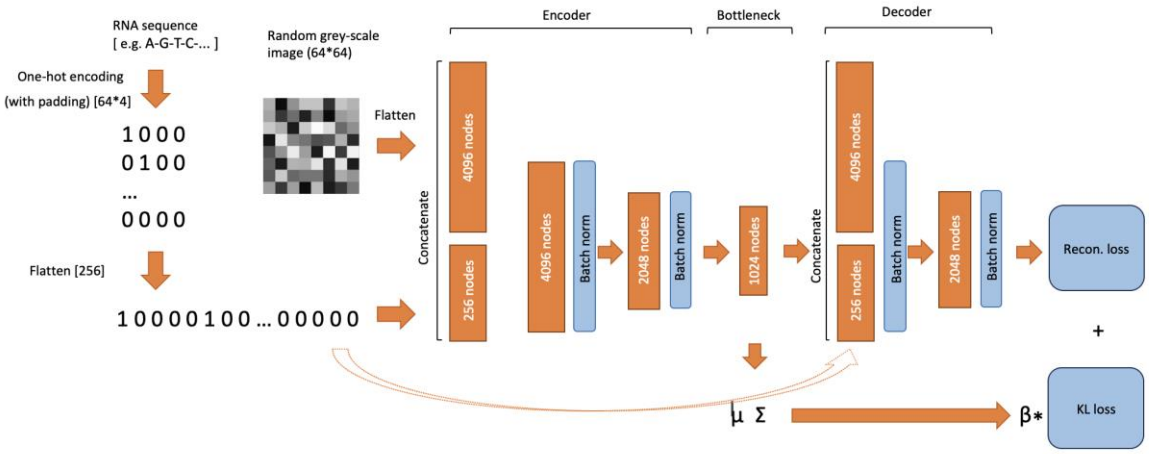
$$L_{MAE} = \sum_{i=0}^{n} \left(dm_{pred}[i] - dm_{true}[i] \right)$$

$$L_{KL} = -\frac{1}{2} \sum_{j}^{n} \left(1 + \log\left(\sigma_j^2\right) - \mu_j^2 - e^{\sigma_j} \right)$$

Where L_{KL}, L_{MAE} are the Kullback-Leibler divergence and the reconstruction losses. σ_j , μ_j are the standard deviation and mean of the $j - th$ dimension of the latent space.

We used mean average error (MAE) as a metric for the algorithm's performance. It is important to note that MAE was calculated only for significant values (the padding values were skipped while calculating the MAE and Loss function). The approximate scheme of the experiment is presented in Figure 2. The architecture was chosen after careful fine-tuning of the model.

Figure 2. The schematic representation of the VAE. The RNA sequences are on-hot encoded into numeric representation and fed together with the random image into the autoencoder. The variational autoencoder consists of 5 hidden layers: the encoder consists of two hidden layers (4096 and 2048 nodes); the bottleneck consists of a layer of 1024 nodes; the decoder consists of two layers (4096+256 and 2048 nodes). The loss function that is used is a combination of the reconstruction loss based on the RMSE and the KL loss.



It is worth mentioning that the distance matrices are supposed to be symmetric. We perform the symmetrization of the predicted distance matrices as a part of postprocessing as follows:

$$dm^{sym} = \frac{dm + dm^t}{2}$$

Indeed, we noticed that if the symmetrization is performed before the calculating of the loss function, the training becomes unstable. If we perform symmetrization before calculating the loss function, a set of values for the predicted distance matrix could satisfy the same value for the symmetric distance matrix. In other words, the sum would matter for the specific element of the symmetric matrix, making the values dm_{ij}, dm_{ji} more flexible by adding an extra degree of freedom. As a result, instability in training occurs.

Results

For the experiment, we trained our model for 1500 epochs. It is worth mentioning that the training occurs for the first 1000 epochs, and the last 500 epochs are not responsible for any performance improvement. Therefore, we used the early stopping method. We stored the model with the best validation MAE and used it for the final prediction. For the training, the batch size was chosen to be 64. This parameter was chosen to make a solution stable, and the learning occurs fast enough. For optimization, we used the Adam optimizer [22]. Regarding the limits of the number of nucleotides, the minimum and maximum lengths of the sequences were chosen for 4 and 64. We used the loss parameter balancing the KL loss and reconstruction loss as the one that gives the best performance and equals 1.1.

We reached the test RMSE error of 3.315Å. Slight overfitting is noticeable (due to the difference in the performance on the validation and train subset). Therefore, we suggest that having more data would only benefit the performance of this algorithm. Please note that we also included the performance results after postprocessing symmetrization. We can see that postprocessing only improves the algorithm's performance. Indeed, the RMSE for the symmetrization is 3.288 Å. In Table 1, the results of the experiment are presented for the train/validation and test subset. The training time for 1500 epochs was around 3 hours (CPU Mac M1 processor). The inference time for one sequence during batch execution occurs to be around 0.001 seconds.

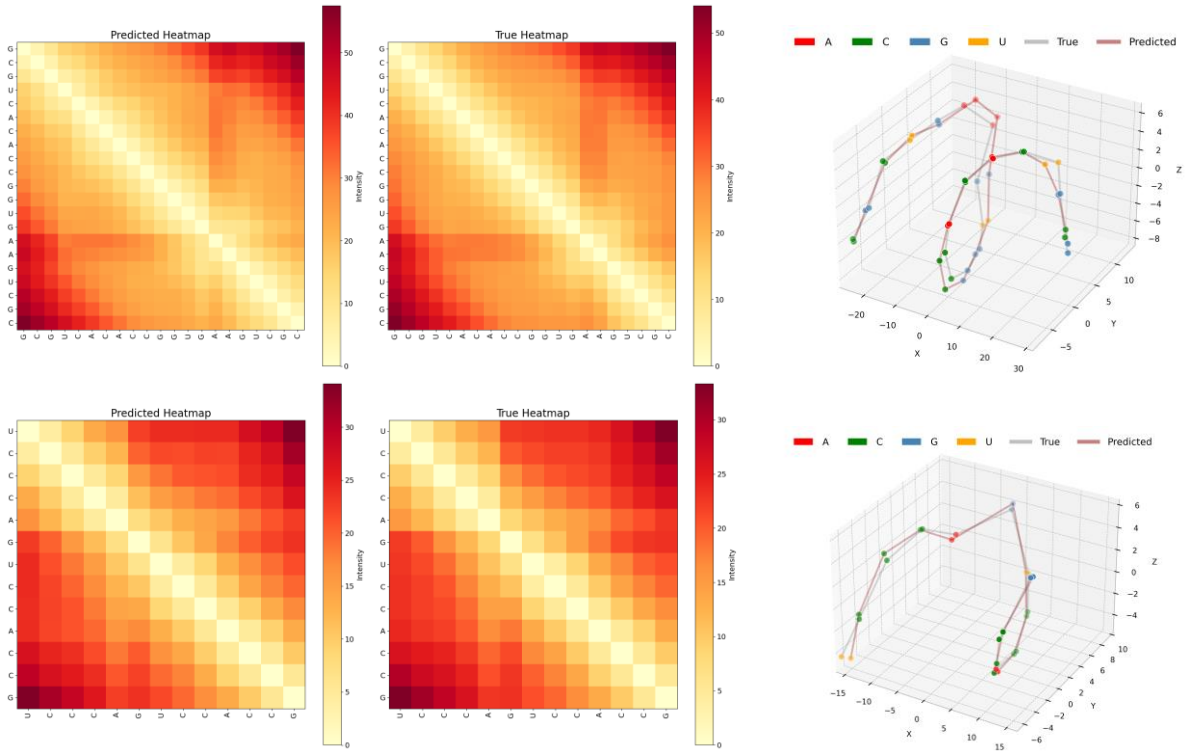
Table 1. The results of the VAE experiment for the Train/Validation/Test datasets. The metrics shown here are MAE, MAE with symmetrization, RMSE, and RMSE with symmetrization.

	Mean Average Error (MAE)	Mean average Error (MAE) after symmetrisation	RMSE	**RMSE (after symmetrisation)**
Train	1.605	-	2.372	-
Validation	2.235	2.235	3.396	**3.376**
Test	2.159	2.142	3.315	**3.288**

Figure 3 displays the computation results for two RNA sequences (for additional examples, please refer to the Supplementary Materials). Within each trio of images, the first two represent the actual and predicted distance matrices, respectively. The third image depicts the three-dimensional structure of the RNA molecule. We employed the distance geometry method to reconstruct the coordinates from the distance matrix, a classic technique proven effective in NMR experimental practices [23]. The challenge here is to find a set of coordinates that optimally fit the given distance constraints. According to [24], this problem can be approached by reformulating it as the task of finding a matrix

$$\min\left|\left|C - G^2\right|\right|, \text{ where } G = -\frac{1}{2}H^T D H$$

Figure 3. Distance matrixes and 3d structure for two RNA sequences (true and predicted) and their 3D structure comparison. The left image represents the Predicted heatmap, the middle image represents the True heatmap, and the right image represents the comparison of 3D structures.

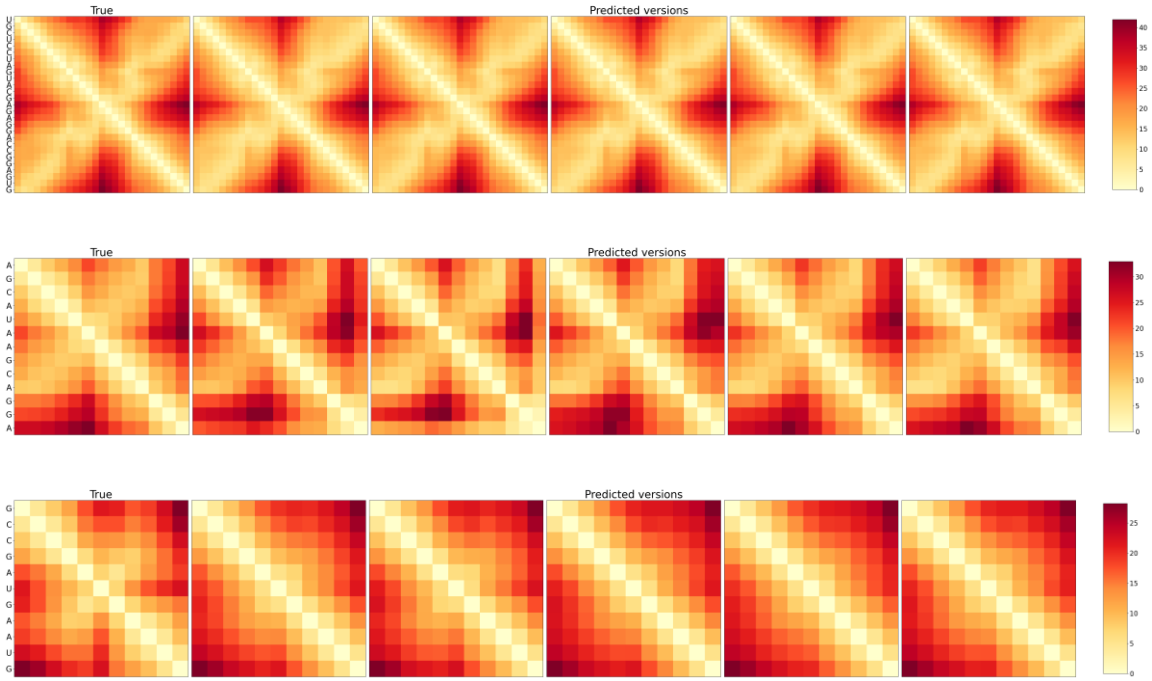


This problem could be solved via computation of spectral decomposition of distance matrix and computation of eigenvalues and eigenvectors. The components of this matrix are the coordinates we need.

In addition to the basic experiments, we decided to check the variational idea of the autoencoder. We generated the distance matrixes five times for several sequences using different noise vectors. In Figure 4, the results of the experiment are presented. We can see that the change in noise vectors does not significantly alter the predicted distance matrix. In other words, the variational factor of the algorithm has minimal influence on the outcome. Let us calculate the average coefficient of variation as a ratio between the standard deviation of the error for the different noise vectors for all the test sequences. Indeed, the ratio is small.

$$Coef.Var. = \frac{\overline{\sigma}}{\mu} = 0.13$$

Figure 4. Generated structures with different noise vectors for three RNA structures. The left images represent the True values, the Five Right images are the predicted structures for different initial noise vectors. The nucleotide labels are shown on the left.



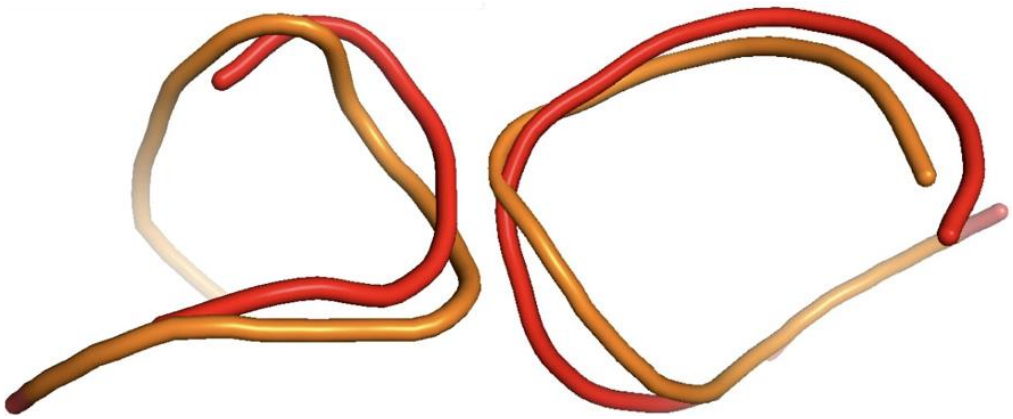
For estimation of the quality of our 3D structure prediction, we compared our algorithm with RhoFold [25], trRosettaRNA [12], and SimRNA [16]. For the simulation in SimRNA and RhoFold, we used standard parameter preset with 10000 cycles. The comparison was provided for the random subset, which was not used in the training (or validation). For RMSD computation between experimental and modelled structures, we used PyMol [26] package. The results are presented in Table 2.

Table 2. The comparison of existing algorithms with the developed algorithms.

	Our algorithm	Rfold	SimRNA	trRosettaRNA
RMSE (Å), less than 25 b.p	**1.917**	3.07	3.1	2.65
RMSE (Å), more than 25 b.p.	**4.018**	5.04	4.97	5.01

In addition, we have checked our neural network for points model from RNA for recovering the entire structure of RNA. In regular conditions, all nucleotides in RNA have flat conformations; for recovering structure, we need only three atoms of nucleotides. We used C1', N9, and C8 atoms to recover the nucleotide plane. For RNA backbone recovery, we used atoms P, O5', C4', C3', C2', O4', and C5'. Also, we added one linear layer in the encoder and decoder. Other steps of the pipeline were identical to the abovementioned steps. During this experiment, we obtained an MAE (Mean Average Error) of 1.8 Å on the test dataset. Therefore, we can infer that our model can be expanded on all atoms in the RNA structure.

Figure 5. Comparison between structure of RNA stem loop 1ZIG and 7KUD PDB (left and right) structures for experimental and predicted structure.



Discussion

The presented architecture of the autoencoder is one of the possible approaches for RNA folding problem-solving. Another potential idea for improving quality prediction could be to use new techniques such as diffusion models [27] or graph neural networks. Also, a very promising path would be adding several RNA-specific features into the model based on the biophysical properties of molecules. For example, as an additional feature, we can consider the secondary structure of RNA [28], the functional class of predicted molecules. As shown in protein cases [2], significant improvement in model quality could be achieved by the addition of evolutionary information [28]. But, instead of a protein folding problem, RNA molecules do not have a good model of nucleotide coevolution and its connection to RNA ternary structure. Therefore, before adding this coevolution information, we must understand the applicability of such a model to our problem. Another way to improve the architecture's performance is to use physically informed Neural Networks [18] to gain additional information about the interaction of atoms.

Another way to improve the performance is to find a feasible way to augment data. Now, we only have a little data for learning large complex models. Modeling new RNA structures by physical-based methods (Molecular Dynamics, Monte Carlo Sampling) could give us good-quality structures but will be slow and computationally expensive. A practical solution could be to use the rotamers library. We can change one or several nucleotides in RNA on another. Their local conformation must be in accordance with the rotamer library. After that, we must perform several MD steps to remove the clash and improve the structure.

Conclusion

We presented a novel method for predicting of RNA tertiary structure using a Variational Autoencoder. Compared with traditional approaches (e.g., Dynamic Simulations [29]), the method uses the complex non-linear relationship in the RNA sequences to perform the prediction. It broadens our understanding of RNA biology and opens new possibilities in therapeutic development. Indeed, knowledge of the structure of RNA sequences could improve the design of RNA-based drugs and the development of RNA vaccines. We achieved the RMSE of approximately 3.3 Angstrom for predicting of the nucleotide positions, which outperformed the competing approaches. The algorithm still has some limitations regarding the specific RNA sequences due to the need for appropriate training data. For some structures, sub-optimal conformations that could vary from the original tertiary structures are found. Diffusion models and physics-induced models paired with larger datasets can enhance the prediction of the tertiary structure even further. The presented algorithm could be used to predict the RNA tertiary structure (the algorithm can be cloned via the link https://github.com/quantori/AutoRNA).

Contributions

M.K. contributed to planning, pre-processing the dataset, discussion, creating models, performing experiments, writing manuscript.

L.U contributed to planning, pre-processing the dataset, discussion, writing the manuscript.

F. Z. contributed to pre-processing the dataset, discussion, creating visualization

I. P. contributed to discussion, creating visualization

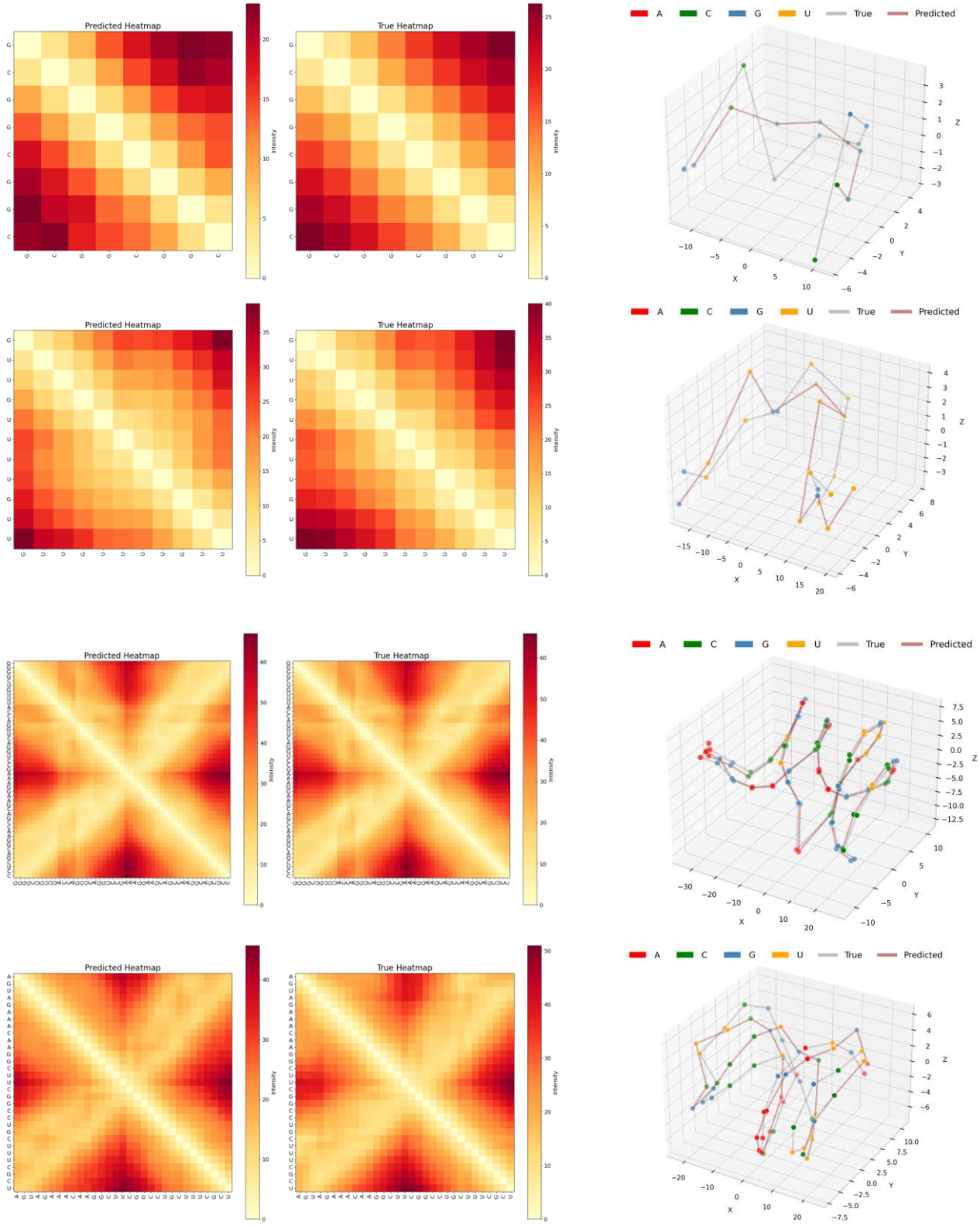
O. K. contributed to discussion, writing a manuscript.

Y. G. contributed to planning, discussion, organizing the research.

Acknowledgements: A. Kasianov, A. Krasnaya, S. Semyonov, A. Safaraleev

Supplementary material

Nucleotide predictions



References

[1] Yu A.M., Choi Y.H., Tu M.J. (2020) RNA drugs and RNA targets for small molecules: principles, progress, and challenges. *Pharmacological Rev.*, 72(4), 862-898.

[2] Jumper J., Evans R., Pritzel A., Green T., Figurnov M., Ronneberger O., Tunyasuvunakool K., Bates R., Žídek A., Potapenko A., Bridgland A., Meyer C., Kohl S.A.A., Ballard A.J., Cowie A., Romera-Paredes B., Nikolov S., Jain R., Adler J., Back T., Petersen S., Reiman D., Clancy E., Zielinski M., Steinegger M., Pacholska M., Berghammer T., Bodenstein S., Silver D., Vinyals O., Senior A.W., Kavukcuoglu K., Kohli P., Hassabis D. (2021) Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873), 583-589.

[3] Burley S.K., Berman H.M., Kleywegt G.J., Markley J.L., Nakamura H., Velankar S. (2017) Protein Data Bank (PDB): The single global macromolecular structure archive. In: Read R.J., Schmidt H. (eds). *Methods in Molecular Biology*. Springer, New York, pp. 627-641. https://doi.org/10.1007/978-1-4939-7000-1_26.

[4] Gillespie J., Mayne M., Jiang M. (2009) RNA folding on the 3D triangular lattice. *BMC Bioinformatics*, 10, 1-17.

[5] Xia Z., Gardner D.P., Gutell R.R., Ren P. (2013) RNA 3D structure prediction by using a coarse-grained model and experimental data. *J. Phys. Chem. B.*, 117(11), 3135-3144.

[6] Tan Y.L., Liu H., Tan Z.J., Shi Y., Zhou P., Lu Z.J. (2022) rsRNASP: A residue-separation-based statistical potential for RNA 3D structure evaluation. *Biophys. J.*, 121(1), 142-156.

[7] Rahmanov S., Kulakovskiy I., Uroshlev L., Makeev V. (2010) Empirical potentials for ion binding in proteins. *Journal of Bioinformatics and Computational Biology*, 8(03), 427-435.

[8] Case D.A., Cheatham III T.E., Darden T., Gohlke H., Luo R., Merz Jr K.M., Woods R.J. (2005) The Amber biomolecular simulation programs. *Journal of Computational Chemistry*, 26(16), 1668-1688.

[9] Sharma S., Ding F., Dokholyan N.V. (2008) iFoldRNA: three-dimensional RNA structure prediction and folding. *Bioinformatics*, 24(17), 1951-1952.

[10] Van Der Spoel D., Lindahl E., Hess B., Groenhof G., Mark A.E., Berendsen H.J. (2005) GROMACS: fast, flexible, and free. *Journal of Computational Chemistry*, 26(16), 1701-1718.

[11] Al-Hashimi H.M., Walter N.G. (2008) RNA dynamics: it is about time. *Current Opinion in Structural Biology*, 18(3), 321-329.

[12] Wang W., Feng C., Han R., Wang Z., Ye L., Du Z., Yang J. (2023) trRosettaRNA: automated prediction of RNA 3D structure with transformer network. *Nature Communications*, 14(1), 7266.

[13] Li J., Chen S.J. (2021) RNA 3D structure prediction using coarse-grained models. *Frontiers in Molecular Biosciences*, 8, 720937.

[14] Watkins A.M., Rangan R., Das R. (2020) FARFAR2: improved de novo rosetta prediction of complex global RNA folds. *Structure*, 28(8), 963-976.

[15] Fu L., Cao Y., Wu J., Peng Q., Nie Q., Xie X. (2022) UFold: fast and accurate RNA secondary structure prediction with deep learning. *Nucleic Acids Research*, 50(3), e14-e14.

[16] Bugnon L.A., Di Persia L., Gerard M., Raad J., Prochetto S., Fenoy E., Milone D.H. (2023) sincFold: end-to-end learning of short-and long-range interactions for RNA folding. *bioRxiv*, 2023-10.

[17] Doersch C. (2016) Tutorial on Variational autoencoders. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1606.05908.

[18] Cuomo S., Di Cola V.S., Giampaolo F., Rozza G., Raissi M., Piccialli F. (2022) Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3), 88.

[19] Asperti A., Trentin M. (2020) Balancing reconstruction error and kullback-leibler divergence in variational autoencoders. *IEEE Access*, 8, 199440-199448.

[20] Dixon W.J. (1953) Processing data for outliers. *Biometrics*, 9(1), 74-89.

[21] Burgess C.P., Higgins I., Pal A., Matthey L., Watters N., Desjardins G., Lerchner A. (2018) Understanding disentangling in beta-VAE. *arXiv.org*. https://arxiv.org/abs/1804.03599.

[22] Kingma D.P., Ba J. (2014) Adam: A method for stochastic optimization. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1412.6980.

[23] Mucherino A., Papajorgji P.J., Pardalos P.M. (2012) Distance geometry: theory, methods, and applications. Springer Science & Business Media.

[24] Trosset M.W. (1998) A new formulation of the nonmetric strain problem in multidimensional scaling. *J. Classif.*, 15(1), 15-35.

[25] Shen T., Chen J., Li Y., Tang L., Huang Y., He J., Wu Z., Xu J., Cheng J. (2022) E2Efold-3D: end-to-end deep learning method for accurate de novo RNA 3D structure prediction. *arXiv preprint arXiv:2207.01586*.

[26] DeLano W.L., Ultsch M.H., Palo Alto C.A. (2002) Pymol: An open-source molecular graphics tool. *CCP4 Newsl. Protein Crystallogr.*, 40(1), 82-92.

[27] Ho J., Jain A., Abbeel P. (2020) Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33, 6840-6851.

[28] Vicens Q., Kieft J.S. (2022) Thoughts on how to think (and talk) about RNA structure. *Proc. Natl. Acad. Sci. U.S.A.*, 119(17), e2112677119.

[29] Šponer J., Bussi G., Krepl M., Banáš P., Bottaro S., Cunha R.A., Gil-Ley A., Pinamonti G., Poblete S., Jurečka P., Walter N.G., Otyepka M. (2018) RNA structural dynamics as captured by molecular simulations: A comprehensive overview. *Chem. Rev.*, 118(8), 4177-4338. https://doi.org/10.1021/acs.chemrev.7b00427.