RNAGEN: A generative adversarial network-based model to generate synthetic RNA sequences to target proteins

Furkan Ozden[1], Sina Barazandeh[2], Dogus Akboga[3], Urartu Ozgur Safak Seker[3,*], and A. Ercument Cicek[2,4,*]

[1] Department of Computer Science, Oxford University, Oxford, UK
[2] Department of Computer Engineering, Bilkent University, Ankara, Turkey
[3] National Nanotechnology Research Center, Bilkent University, Ankara, Turkey
[4] Computational Biology Department, Carnegie Mellon University, Pittsburgh, PA
*{urartu@bilkent.edu.tr, cicek@cs.bilkent.edu.tr}

Abstract. RNA - protein binding plays an important role in regulating protein activity by affecting localization and stability. While proteins are usually targeted via small molecules or other proteins, easy-to-design and synthesize small RNAs are a rather unexplored and promising venue. The problem is the lack of methods to generate RNA molecules that have the potential to bind to certain proteins. Here, we propose a method based on generative adversarial networks (GAN) that learn to generate short RNA sequences with natural RNA-like properties such as secondary structure and free energy. Using an optimization technique, we fine-tune these sequences to have them bind to a target protein. We use RNA-protein binding prediction models from the literature to guide the model. We show that even if there is no available guide model trained specifically for the target protein, we can use models trained for similar proteins, such as proteins from the same family, to successfully generate a binding RNA molecule to the target protein. Using this approach, we generated piRNAs that are tailored to bind to SOX2 protein using models trained for its relative (SOX15, SOX14, and SOX7) and experimentally validated *in vitro* that the top-2 molecules we generated specifically bind to SOX2.

Keywords: Generative Models · RNA design · Deep Learning · RNA-Protein Binding

1 Introduction

RNA vaccines came under the spotlight with the promise to end the COVID-19 pandemic as a modular, effective, and secure option for immunization. While this success has proven the effectiveness of RNA molecules as an active means of disease intervention rather than being a passive drug target [24], RNA-based therapeutics are in their infancy [5]. The current therapeutic landscape heavily relies on small molecule drugs [13] which target ∼4% of all human proteins [48] and only ∼25% of deemed druggable proteins [46]. Utilizing RNA to target proteins has a big potential to bridge this gap.

RNA binding proteins (RBP) are abundant in the human body, and they predominantly interact with non-coding RNAs (ncRNAs). Consequently, diseases associated with RBPs exhibit distinct phenotypes based on the specific types of RNA to which these RBPs bind [20]. This specificity is an important aspect in minimizing off-target effects. Furthermore, the capacity for rapid modifications in RNA sequences presents opportunities for personalized treatment approaches and allows for adaptation to evolving pathogens. These advantages further emphasize the potential of RNA-based drugs in the realm of therapeutic interventions. Here, we focus specifically on the piwi-protein-binding family of RNAs, also known as piRNAs, which have shown promise as novel biomarkers for early diagnosis of breast cancer [32] and as therapeutic targets for personalized medicine [53] but have not been used as drugs to the best of our knowledge. They have relatively short sequences (24 to 32 bp), which makes them easier to generate synthetically and to specialize to target certain proteins.

Various techniques have been proposed for engineering and *de novo* RNA sequences, including CRISPR-based methods, structural designs, and fully computational methods. The conventional approach involves tackling the inverse folding problem, where a target secondary structure is provided, and a corresponding sequence is generated. Several algorithms have been employed to design RNA sequences using this method, including both target-specific and general methods [31] [11] [16] [47] [52]. RNAifold and MioRNAifold use a constraint programming-based approach to search for the RNA with the Minimum Free Energy (MFE) leading to the target structure [19] [40]. Other studies use evolutionary algorithms along with constraint-based approaches for designing the target sequences [15] [39]. Similarly, there are models that use a target tertiary structure of

the RNA for obtaining the original sequence [51]. On an orthogonal track, a CRISPR-based method is used for designing guide RNAs (gRNA) [35]. It involves editing the RNA sequence using CRISPR and applying various algorithms and rules to select the best versions. A fully in-silico approach is to use a 3D motif library drawn from all unique, publicly deposited crystallographic RNA structures and uses an efficient algorithm to discover combinations of these motifs and helices that solve the RNA motif path-finding problem [54]. When this 3D structural design is completed, they are filled in with sequences that best match the target secondary structure and minimize alternative secondary structures. Although these studies have been successful in generating or modifying RNA sequences to obtain a desired target, the utilized approaches are not able to span the entire space of piRNA sequences due to the limitations of their methodologies. Additionally, these models typically necessitate either an input RNA sequence for a mutation-based optimization or a provided 2D or 3D structure to facilitate sequence design.

More recent approaches benefit from deep learning for designing biological sequences, including DNA [6] and RNA sequences [26] [10]. Huang et al. use a graph neural network for imputation of the RNA sequences [26] as a way to generate the missing sequences obtained from the RNA sequencing devices. Chuai et al. combine the use of a deep learning model with the CRISPR method for designing the RNA sequences [10]. Similar to some older studies, this approach requires a starting RNA sequence and is not capable of generating truly novel RNA sequences. Furthermore, these models do not offer additional control over the functional attributes of the designed sequences, such as their capacity to bind to specific proteins.

We propose RNAGEN, the first computational model based on a generative adversarial network (GAN) architecture for generating realistic piRNA sequences. This architecture is matched with an optimization procedure to tailor the generated sequence for binding to a target protein. We use differentiable models such as DeepBind [2] that predict a binding score to a target protein for an input RNA sequence. Our pipeline can also design sequences for proteins for which no such model exists if there are available models for related proteins, such as other proteins from the same family. RNAGEN does not require a known target secondary or tertiary structure, nor does it require any starting sequence for optimization. Our model performs the generation and the optimization independently, making it suitable for any optimization task [44] depending on the availability of a differentiable prediction model for the selected task. We show the generated sequences resemble properties of natural piRNAs with respect to various metrics, and optimized sequences have a substantially larger affinity to bind to the target protein. We synthetically generate piRNA sequences that are optimized to bind to SOX2 protein, using DeepBind models trained for other proteins from the SOX family. We conduct in vitro experiments and show that RNAGEN-generated piRNA sequences display binding specificity to SOX2. We think RNAGEN is an efficient technique for designing RNA sequences as candidate drugs and opens up a new promising venue for RNA-based therapeutics.

2 Results

2.1 Overview of RNAGEN

The RNAGEN model is a deep generative adversarial network (GAN) that learns to generate piRNA sequences with similar characteristics to the natural ones. This model is a novel version of the WGAN-GP architecture [23] for one-hot encoded RNA sequences. The model provides improved training over the original Convolutional GAN (DCGAN) models [23] [43] and is less prone to overfitting than the original WGAN architecture [3].

Figure 1A shows the overview of the WGAN architecture. The input is a noise vector. The generator and the critic are convolutional neural networks trained together. The generator upsamples the input vector using transpose-convolutions to generate piRNA sequences, whereas the critic uses convolutions and dense layers to score the actual and generated samples. The optimization pipeline is shown in Figure 1B where we select three proteins of the same family that are closest to the target protein in the Prot2Vec [4] space and use the distance as weight and the DeepBind [2] predictions as scores, the input vector is learned to generate piRNA sequences with higher binding affinity to the relative proteins and therefore the target protein itself.

2.2 Generated sequences maintain their distance to the natural piRNAs

The WGAN architecture described above (and also in detail in Section 4) learns to generate piRNA sequences that resemble the natural piRNA sequences. We compared the distances between synthetic and natural se-

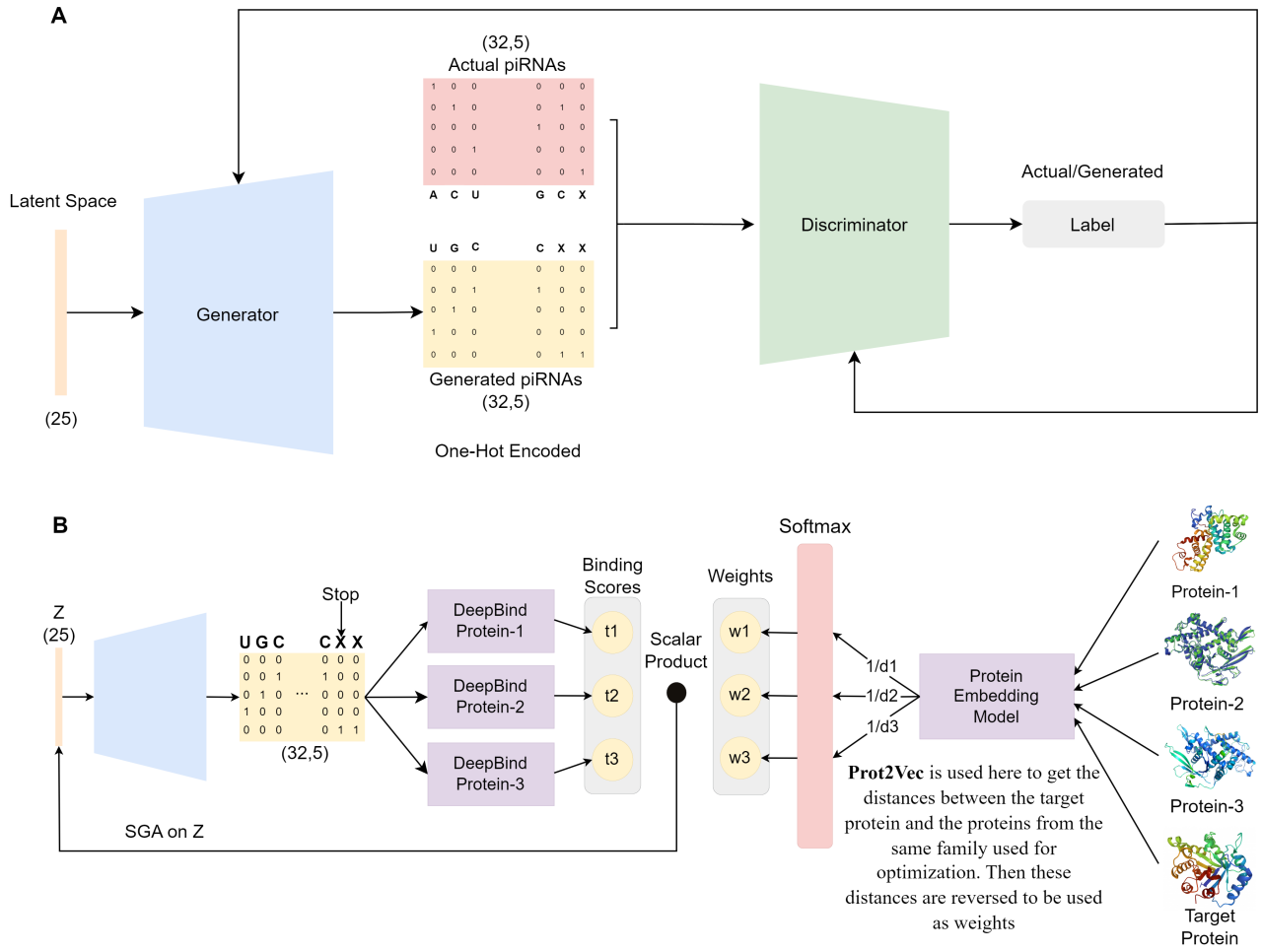

Fig. 1. Training and Optimization pipeline. A shows the training of the GAN model. The GAN consists of a generator and a critic. The one-hot encoded natural sequences are used to train the model, and the generator also outputs padded one-hot sequences. The sequence is derived from the one-hot encoding, and the conversion stops with the first padding alphabet. **B** is the optimization pipeline, where the generator G is trained on real piRNA sequences. We utilized the Prot2Vec model [4] to find proteins similar to our target protein for which pre-trained DeepBind [2] models exist. The Prot2Vec model is a deep learning model that generates numerical representations of protein sequences, enabling similarity comparisons and downstream analysis. RNAGEN learns latent vectors z that lead to the generation of optimized piRNA sequences with improved binding scores to the target protein. Our model uses inverse distances (d) of relative proteins to the target protein as a weighting tool by which it adjusts the relative improvements to the latent vector z. We multiply these weights with the respective binding score predictions from the pre-trained DeepBind models. The result of the scalar product is then treated as a hint score whereby providing a differentiable score function for z to optimize for.

quences. Due to the variable length of the piRNA sequences, we selected the Levenshtein distance as the distance metric. Levenshtein distance is the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one word into the other [34]. To evaluate the distance between the set of generated and random sequences in relation to the natural dataset, we found the distance of each generated or random sequence to the closest sequence in the natural samples in terms of Levenshtein distance. Then we employ these sets of distances to represent the distribution of distances within these sets. To enable a comparison with the natural dataset, we performed a similar analysis on the natural samples by finding the smallest non-zero distance between each natural sample and the entire set of natural piRNA sequences. The distributions shown in Figure 2A demonstrate that the generated sequences exhibit a nearly identical distribution of distances compared to the natural sequences, whereas randomly generated samples displayed a distinct distribution. Furthermore, the average distance of the random sequences to the natural sequences is higher than the generated ones, indicating that the generative model has effectively learned the contextual characteristics of the sequences. We used the Mann-Whitney U-test here, which is a non-parametric statistical test used to compare differences between two independent sets of samples and is usually preferred over the t-test when the data is following non-normal distributions [38]. The two-sided Mann-Whitney U-test [38] results in a p-value of smaller than 0.01 for random sequences and a p-value of 0.18 for the generated sequences which shows that they are not significantly different than the natural sequences. We calculated the effect size in our experiments using the non-parametric delta statistics (Cliff's delta) [12] and the 95% confidence interval [8] of the medians for the sets of values. The calculated effect size and confidence interval for the sets of random and natural sequences are 0.82 and (3,3), respectively, indicating a large effect. The values using the same methods for the sets of generated and natural sequences are 0.01 and (0,0), respectively. These results support the **p**-value in showing that the difference between the natural and generated sequences is not statistically significant.

2.3 GC Content of the natural and generated sequences are similar

We investigated the GC content of the generated sequences. We calculated the GC content of 2048 generated sequences and compared the distribution to that of 2048 natural RNA sequences. The average length of both sets of sequences was 28 bp. The results in Figure 2B show that the RNAGEN generated sequences with similar GC content compared to the natural piRNAs. While the average GC content of the entire genome was 59.1%, the GC content of the natural sequences averaged just above 50%. Although the Mann-Whitney U-test showed a significant difference ($p < 0.01$) between the distributions of GC content in the natural and generated sequences, the average GC content was similar, and the GC content of the generated sequences fell within the range observed in the natural ones. The effect size (Cliff's delta) was -0.21, indicating a small effect, and the 95% confidence interval of the medians was (-0.034, -0.0224). Since we needed to input the GC content of random sequences while generating them, we did not include the GC content of random sequences in this analysis as it would not contribute to the validation process. The GC content of random sequences using the RNA alphabet is always 50%.

2.4 Distribution of the minimum free energy of the generated is similar to that of natural ones

We investigated the MFE of the generated piRNA sequences. MFE of an RNA sequence is the energy of the secondary structure that contributes a minimum of free energy [55]. We used the ViennaRNA package [36] to measure the MFE of the RNA sequences, and the measurements using the Nupack package [17] also yielded similar results. In Figure 3, we compared the MFE distributions of 19200 generated sequences and the entire set of natural piRNA sequences used for training and validation. The difference in distributions was statistically insignificant (Mann-Whitney U-test: $p > 0.01$, 95% confidence interval of the medians: (0.09, 0.39), Cliff's delta effect size: 0.06). Also, we randomly generated a matching number of RNA sequences and compared their MFE distribution with natural and generated sequences. The MFE distribution of random sequences was substantially different from the counterparts (Mann-Whitney U-test: $p < 0.01$, 95% confidence interval for medians: (-1.69, -1.30), Cliff's delta effect size: -0.31). This shows that RNAGAN is able to mimic natural piRNAs in terms of their minimum free energy.

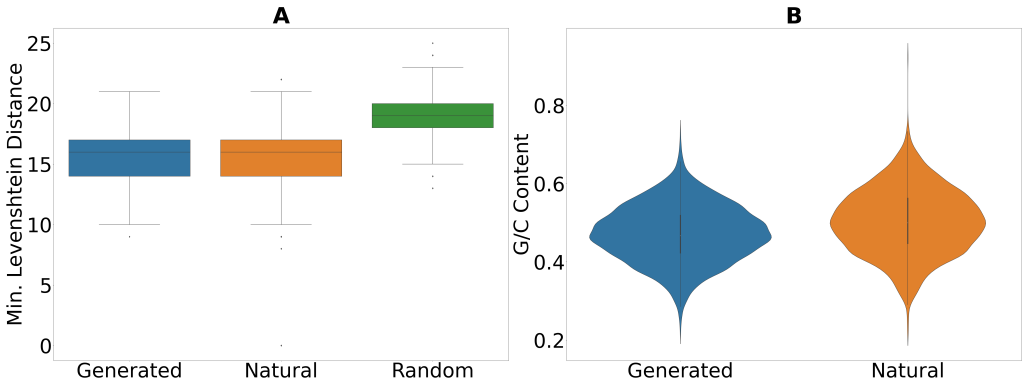

Fig. 2. G/C content and Levenshtein distance distributions. A The Levenshtein distance of the generated and random sequences to the closest natural sequence is measured, and the distributions of the distances are shown here. We expected the sequences not to be very similar or very distant from the natural set of sequences. To measure this for the actual sequences, the distance to the closest member with a non-zero distance is considered. We use the Mann-Whitney U-test for statistical tests, and as shown here, the distance is maintained (Mann-Whitney U-test: $p = 0.18$, 95% confidence interval for medians: (0, 0), Cliff's delta effect size: 0.01). The p value for the random sequences was smaller than 0.01 (95% confidence interval for medians: (3, 3), Cliff's delta effect size: 0.82). The size of the random and generated sequences are both 2048, and all the 50,397 natural sequences are used here. The box plot characterizes a sample using the 25th, 50th, and 75th percentiles—also known as the lower quartile (Q1), median (m or Q2), and upper quartile (Q3)—and the interquartile range (IQR = Q3 - Q1), which covers the central 50% of the data. The whiskers extend to a maximum of 1.5 times IQR beyond the box. Any data points beyond the whiskers are considered outliers and plotted individually as points. **B** The GC content of the 50,397 natural and 2048 generated sequences are shown with similar mean values. The range of the GC content of the generated one remains in the range of natural sequences while the statistical test does not show similar distributions (Mann-Whitney U-test: $p < 0.01$, 95% confidence interval for medians: (-0.034, -0.0224), Cliff's delta effect size: -0.21).

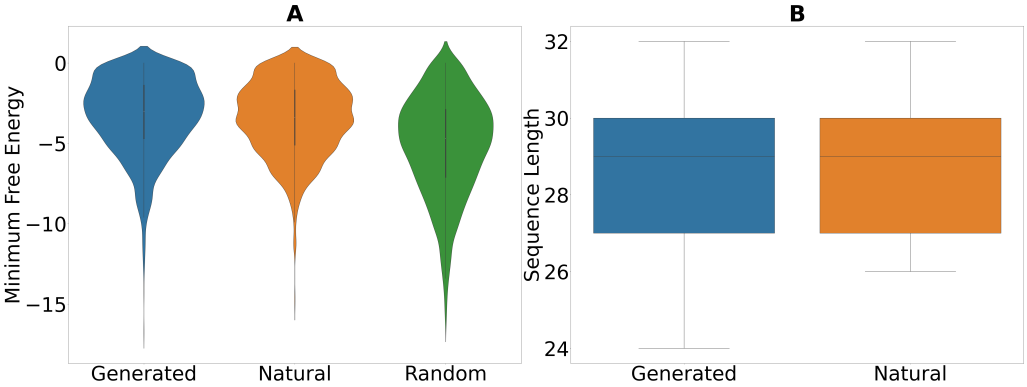

Fig. 3. Minimum Free Energy distributions. A shows the distribution of the MFE in the generated, random, and natural piRNA sequences. The random sequences are generated based on the average GC content of the reference genome. The distribution of the MFE in 2048 randomly generated sequences is skewed (Mann-Whitney U-test: $p < 0.01$, 95% confidence interval for medians: (-1.69, -1.30), Cliff's delta effect size: -0.31) while the 50,379 natural and 19,200 generated sequences are considerably similar in distribution (Mann-Whitney U-test: $p > 0.01$, 95% confidence interval for medians: (0.09, 0.39), Cliff's delta effect size: 0.06). **B** The distributions of the length of the generated and natural sequences are shown to be similar, with almost identical averages of 28.6 and 28.5, respectively. The length of the sequences affects the MFE, and the results show that the generative model learns to maintain the length as well as the MFE. The box plot characterizes a sample using the 25th, 50th, and 75th percentiles—also known as the lower quartile (Q1), median (m or Q2), and upper quartile (Q3)—and the interquartile range (IQR = Q3 - Q1), which covers the central 50% of the data. The whiskers extend to a maximum of 1.5 times IQR beyond the box. Any data points beyond the whiskers are considered outliers and plotted individually as points.

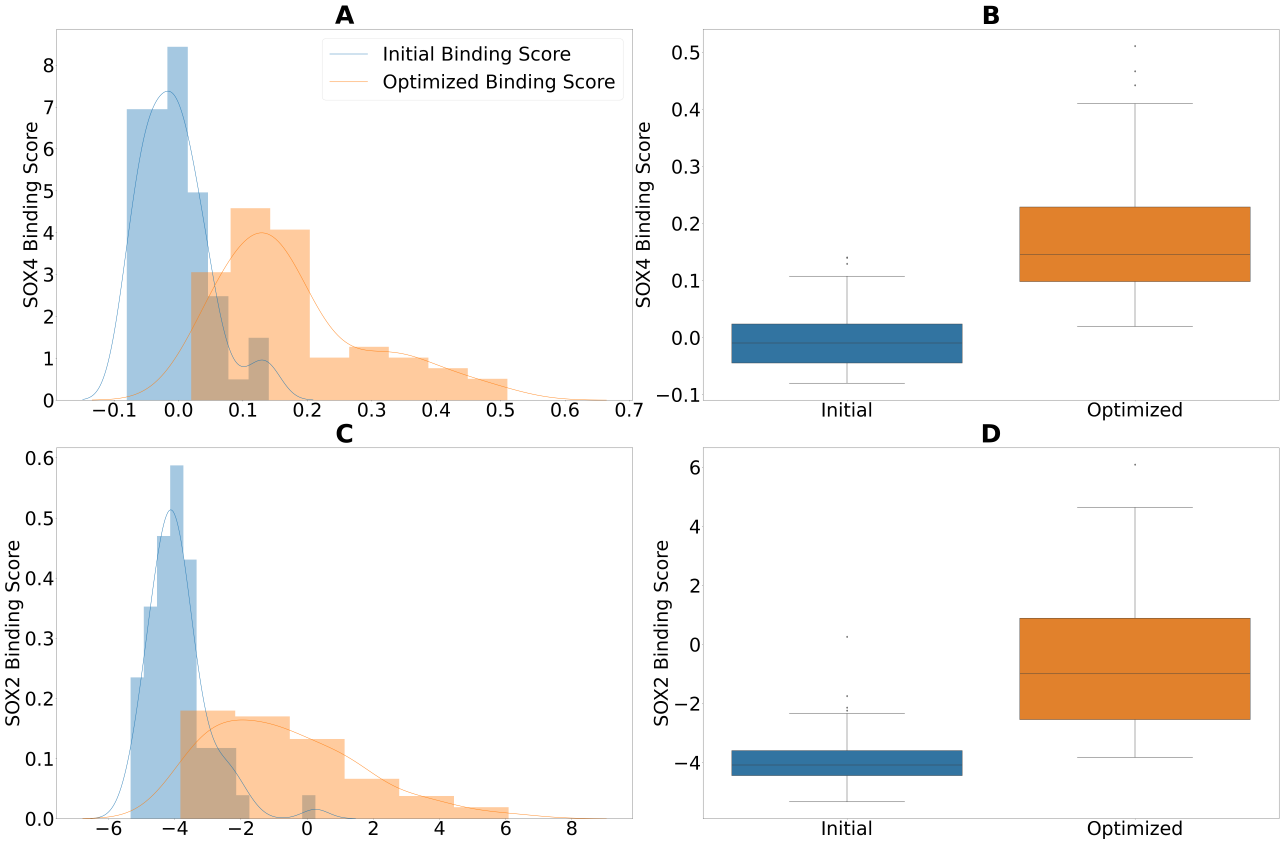

Fig. 4. Predicted binding scores before and after optimization. A,**B** Show the distribution of predicted binding scores of the initially generated and optimized piRNA to the SOX4 protein using DeepBind models of the SOX8, SOX7, and SOX10 proteins. As shown in both plots, the predicted binding scores increased substantially, and there was a three-fold increase in the maximum predicted binding score after optimization. **C**,**D** Show the initial and optimized predicted binding scores of the generated piRNA sequences to the SOX2 protein using SOX15, SOX14, and SOX7 proteins as proxy proteins for optimization. In this case, not only the maximum predicted binding score is an order of magnitude larger, the highest score reached the considerably high score of 6.09. Box plots both in **B** and **D** characterizes a sample using the 25th, 50th, and 75th percentiles—also known as the lower quartile (Q1), median (m or Q2), and upper quartile (Q3)—and the interquartile range (IQR = Q3 - Q1), which covers the central 50% of the data. The whiskers extend to a maximum of 1.5 times IQR beyond the box. Any data points beyond the whiskers are considered outliers and plotted individually as points.

2.5 Electrophoretic Mobility Shift Assay (EMSA) confirms binding of two of the optimized piRNA sequences with highest binding specificities

We used our proposed architecture to generate piRNA sequences, specifically designed to have high binding scores to human transcription factor SOX2. Figure 4 shows binding scores predicted by the DeepBind model for the SOX4 (see Figure 4A and 4B) and SOX2 (see Figure 4C and 4D) proteins, of the generated piRNA sequences before and after performing optimization for each of these proteins separately using our model. To perform the optimization, we selected SOX15, SOX14, and SOX7 proteins as the closest relatives for SOX2, and for SOX4, we selected SOX8, SOX7, and SOX10 proteins. We assumed that there are no DeepBind models for SOX2 and SOX4 proteins, while DeepBind models are available for the selected relatives. The distance between SOX2 and the selected relative proteins in the Prot2Vec space is 40.81, 44.86, and 54.14, respectively. The distance of SOX4 from its relatives is 89.08, 95.59, and 106.18 in the same order. Although this model is more useful for proteins without DeepBind prediction models available, here we select proteins (SOX2, SOX4) for which DeepBind models exist to be able to validate the optimization results. In the optimization procedure, 64 piRNAs are generated using the trained generator, and then we optimize the generator's input latent vector using gradient ascent to maximize the predicted binding score to the target proteins. We validate the optimization using *in vitro* experiments on the SOX2 protein. We observe that the optimization procedure yields piRNA sequences that have high predicted binding scores against SOX2. The maximum predicted binding score obtained is 6.09, which demonstrates a well-binding preference as mentioned in [2]. The optimization does not yield similarly high predicted binding scores for SOX4, and the distances of the closest relatives to this protein are also considerably higher than those of SOX2. Two optimized piRNA sequences with maximum predicted binding scores to SOX2 (dubbed RNA aptamer 1 and RNA aptamer 2) were studied via *in vitro* validation experiments. We used electrophoretic mobility shift assays (EMSAs) [25]. They analyze interactions between proteins and nucleic acids. The assays use polyacrylamide gel electrophoresis (PAGE) under non-denaturing conditions and buffers at or near neutral pH. The migration of free and bound biomolecules in an electric field is measured to determine the affinity and specificity of the interaction [45, 49, 50]. Here, we used an agarose gel electrophoresis technology that has been developed to allow for short run times while simultaneously yielding high band resolution and avoiding short-RNA aptamers from degrading, and detecting RNA-bound protein from free RNA with a nucleic acid dye, which is safer than conventional radioactive labeling [42]. *In vitro* binding reactions were visualized using agarose gel-based rapid EMSA, which was performed to verify the binding between obtained RNA aptamers and SOX2 protein (Figure 5). The arrow in Figure 5, indicates the *in vitro* binding of piRNA sequences and purified SOX2 protein, demonstrating saturation as the RNA concentration increases. Unbound RNA, being molecularly smaller than the RNA-protein complex, moves faster in the agarose gel with applied voltage. The gradient of this movement is marked with a bracket. The bands indicated by the arrow suggest that piRNAs, RNA aptamer 1 and RNA aptamer 2, bind to purified SOX2. See Section 4.4 and Figure 6 for experimental details and the pipeline.

3 Discussion

We introduce RNAGEN, the first framework for generating optimized piRNA sequences with desired binding properties using indirect optimization. Our approach enables us to generate sequences with similar attributes to the natural piRNA samples without the need for exploring intractable search space of RNA sequences that have lengths of up to 32 nucleotides. Our model can also generate sequences of variable lengths, further increasing the possible space of generated sequences.

RNAGEN is a robust framework for synthetic piRNA engineering and is applicable to numerous applications for genetic circuit design. The model's components consist of a GAN adopted to be trained on the human natural 5' piRNA dataset and state-of-the-art protein-binding prediction models as well as protein embedding models. Our model can potentially optimize sequences for any objective function as long as a differentiable prediction model exists to predict the target metric for a piRNA sequence. Similarly, the model can work with classification models to generate sequences of a particular class if the prediction model is replaced with such a classifier.

Our model does not rely on the availability of predictions for the binding scores of the generated piRNA sequences and can optimize piRNAs for the target protein using proteins similar to the target protein in the Prot2Vec [4] space. For this purpose, proteins of the same family for which RNA-protein binding prediction

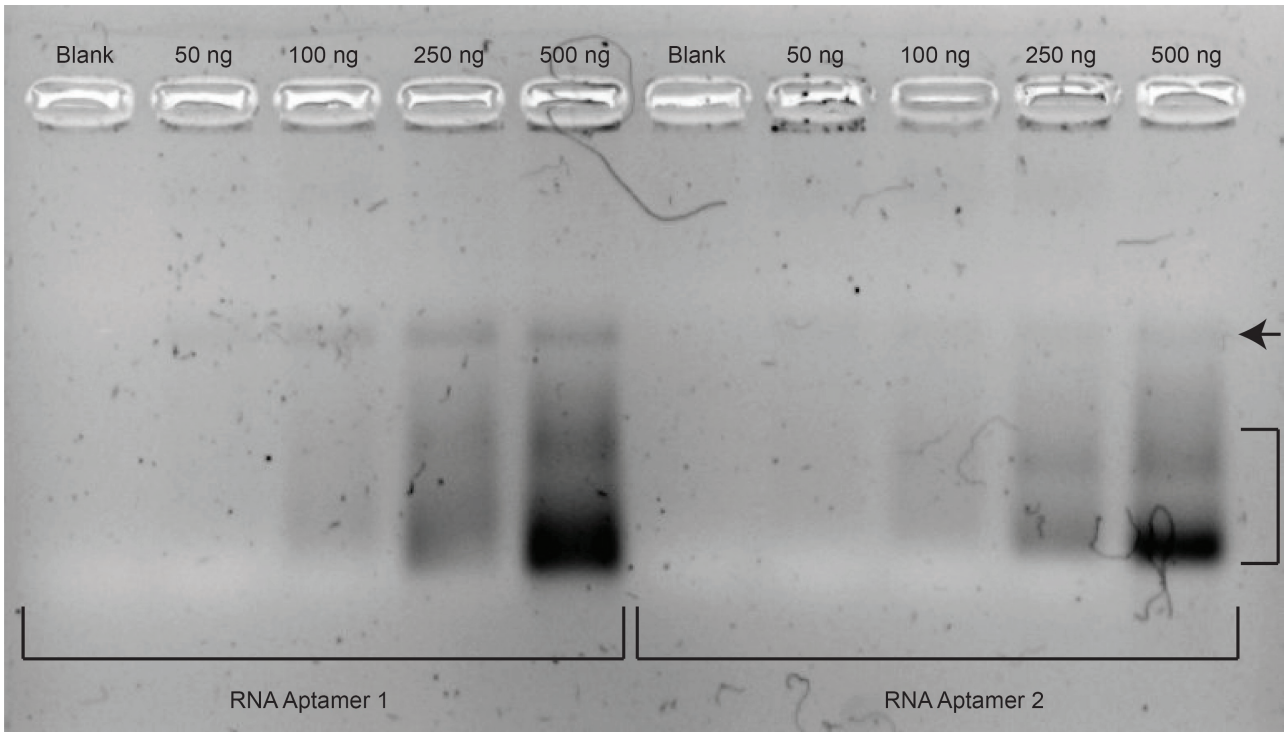

Fig. 5. Rapid agarose gel electrophoresis for electrophoretic mobility shift assay. Binding reactions with a set amount of purified SOX2 protein (2.5 μg) and a concentration gradient of RNA aptamer (The amounts of RNA are equivalent to 430 nM - 4.3 μM for RNA aptamer 1 and 460 nM - 4.6 μM for RNA aptamer 2) were loaded onto the gel. The saturation of RNA-protein binding is indicated by an arrow. The shift of RNA aptamer-protein binding to free RNA is indicated by a curly bracket.

models exist are selected. By increasing the weighted average binding score for these three proteins, the average predicted binding score of the generated piRNA sequences to the target protein is expected to increase.

While RNAGEN is capable of generating and optimizing sequences with favorable binding scores, the optimization process does not consistently yield sufficiently high predicted binding scores. One potential explanation for this outcome is the constraint imposed by the predictor model (DeepBind), which is not perfectly accurate. Moreover, the selection of relative proteins for the optimization process is based on their similarity in the Prot2Vec space, which may not align with their similarity in the DeepBind models' space. Another limitation of RNAGEN is its reliance on available DeepBind models to select proteins similar to the target protein, disregarding the absence of DeepBind models for certain proteins. Consequently, the top three selected proteins might lack the necessary similarity to facilitate meaningful optimization. Thus, when using proteins selected from the Prot2Vec space, the enhancement in the predicted binding score for some proteins may be relatively less significant, as seen in Figure 4A and 4B for the SOX4 protein. Additionally, in rare cases, optimization may lead to decreased predicted binding scores. However, given the flexibility of the optimization steps, if no improvement is achieved after multiple iterations, the best initial predicted binding score can be utilized.

We expect to see wide use of data-driven sequences generated using deep learning methods in the near future. Generating RNA utilizing this approach is practical and does not require massive computational resources. While we focus on piRNA, it is straightforward to generalize the framework to other families of RNA sequences or specific regulatory elements. It is also possible to use other generative models, such as transformers or diffusion models, for more complicated tasks or families of longer sequences. Generating piRNA sequences with specific binding targets enables researchers to conduct experiments in a much shorter time, which can be particularly beneficial in RNA-based drug design.

Our *in vitro* experiment, rapid-EMSA, indicates a promising specificity of the top-2 generated piRNA sequences towards the target, SOX2 protein. Although not quantified, the observed increasing band intensity at protein-RNA binding and saturation of this binding provides preliminary evidence of binding specificity (Figure 5). This result is further supported by the lack of a saturation gradient when the top-2 generated piRNA sequences were tested against Bovine Serum Albumin (BSA), a protein not targeted by our piRNAs (Figure S5). However, to further substantiate these findings, it is essential to quantify the binding rate and the mode of the affinity of these piRNAs to their target protein.

A combination of techniques can be used to elucidate the binding dynamics of our generated piRNAs. Surface Plasmon Resonance (SPR) can monitor binding kinetics in real-time, providing a practical understanding of the interaction rates [28]. To further comprehend the in vitro binding affinity of piRNAs to their target protein, an RNA pull-down technique that allows for the isolation and subsequent identification of proteins that specifically bind to the piRNAs can be employed [9]. Lastly, it is essential to utilize a combination of techniques for the reliable characterization of RNA-target interactions. For instance, the use of isothermal titration calorimetry (ITC), mass spectrometry (MS), and nuclear magnetic resonance (NMR) could provide comprehensive insights into again the binding affinity, selectivity, and mechanism of the generated piRNAs toward their target protein [7]. These methods could offer a more applicative view of the binding dynamics of our piRNAs.

RNA molecules generated with RNAGEN that display specific protein targeting could be utilized as aptamers in RNA-based therapeutics or as sensory elements in molecular sensors for tracking diseases and identifying cellular phases [14, 41]. For instance, these RNA molecules generated with this technique can be enhanced by increasing the stability of their secondary structures in their active conformation and be delivered in synthetic mRNA devices to distinguish cell types [29]. Our top two piRNAs targeting SOX2 protein could be utilized again in synthetic genetic circuits to track SOX2 trajectories and monitor reprogramming efficiency during induced pluripotent stem cell (iPSC) generation [27]. Furthermore, the generated RNA molecules could be utilized to control gene expression by targeting transcription factors in metabolic engineering [30].

4 Methods

4.1 Data set

Our proposed method needs an RNA sequence data set to train the GAN. Even though any computationally suitable RNA type can be modeled, in this work, we focused on piRNAs due to their shorter length, as mentioned before. This characteristic makes it easier for the GAN to learn the underlying sequence patterns effectively. To train the generator model, we used the DASHR2-GEO piRNA database [33]. The dataset contains $50,397$ natural piRNA sequences of lengths varying between 26 and 32 bps. The distribution of the length of these sequences is shown in Figure 3.B and compared to those of synthetic piRNA sequences.

4.2 Experimental Setup

We encode the data using one-hot encoding, and the dimension of each RNA is $(32, 5)$, where 32 is the maximum size, as mentioned, and 5 is the length of the one-hot vector. The batch size during the training and optimization is 64, and we randomly divide the dataset into training and validation sets where the validation set consists of 12.5% of the sequences. The RNA alphabet in one-hot encoded as follows: A:(1,0,0,0,0), C:(0,1,0,0,0), G:(0,0,1,0,0), U:(0,0,0,1,0), X:(0,0,0,0,1) where X is the padding character used when the sequence is shorter than 32 bps. This padding is necessary because the input of the discriminator (critic) as a convolutional neural network has to be in fixed dimensions, and the sequences naturally vary in length. In addition, to generate truly random sequences, we select a random length between 26 and 32 nucleotides and then select each nucleotide randomly from the RNA alphabet with the probabilities according to the GC content of the reference piRNA dataset. As mentioned above, we used Cliff's delta statistics [12] to measure the effect size in our statistical tests and the method mentioned in [8] to calculate the confidence interval. The model is implemented using the Tensorflow 1 [1] library and in Python. We train and optimize the model on a Super-Micro Super-Server 4029GP-TRT with 2 Intel Xeon Gold 6140 Processors (2.3 GHz, 24.75 M cache), 256 GB RAM, 6 NVIDIA GeForce RTX 2080 Ti GPUs (11 GB, 352 bit), and 2 NVIDIA TITAN RTX GPUs (24 GB, 384 bit). We only use a single TITAN RTX GPU. The training takes around 86 hours for 200,000 epochs, and the optimization takes roughly 5 minutes for 3000 iterations. The code is available at https://github.com/ciceklab/RNAGEN.

4.3 Problem Formulation

In this work, we assume that only the sequence information of the target protein is known, and no other computational tool is available to assess the binding score of any given RNA sequence to the target protein. If there is such a computational tool available, the availability of such a computational tool requires more information than the sequence information alone, and then the optimization procedure is more straightforward. Let P_t be the sequence information of the target protein for which we want to design a piRNA to have a high binding preference to it. We assume that a cohort of binding score assessment models for well-known proteins (e.g., DeepBind models) are available. Let the family of assessment functions be $\{F_i(R)\}$ where R is a variable representing any possible input RNA sequence and the corresponding protein sequence information for the function $F_i(R)$ be P_i. Since $\{F_i(R)\}$ does not contain $F_t(R)$ (i.e., the assessment function for target protein), we imagine a latent function $RB_t(R)$ which returns a ground truth real binding score of any input RNA sequence to the target protein. Then, our goal is to find an optimal $R = R^*$ such that $\arg\max RB_t(R) = R^*$. This is without knowing any information on the latent function $RB_t(R)$ while using P_t, $\{F_i(R)\}$ and $\{P_i\}$

4.4 Model

The architecture shown in Figure 1 is composed of (i) a trained generator; (ii) the required number of trained property assessment models, and (iii) a trained protein embedding model. The model requires two optimization procedures. The first required optimization procedure where a generative model (see Figure 1A) consisting of a generator G and a critic C is trained. In this architecture, a GAN is used to capture the latent piRNA distribution and limit the optimization space of the random noise Z (i.e., the second required optimization procedure shown in Figure 1B) with the realistic piRNA subspace. The employed approach provides a realistic optimization on the latent noise Z since each time it passes through the generator, the resulting generated piRNA sequence must belong to the subspace spanned by the trained generator. The purpose of the overall architecture is to find an optimal $Z = Z^*$ (i.e., latent noise for the generator), such that $G(Z^*)$ outputs the desired RNA sequence with maximum binding preference to the targeted protein, where G is the generator function.

Generative Model Architecture The GAN used here is based on the original Wasserstein GAN (WGAN) [3] model that is an improved version of the Deep Convolutional GAN (DCGAN) model [43] with the Wasserstein loss [18]. However, there are challenges to training a Wasserstein GAN in a stable manner. To ensure stable training of the network, we use an improved training procedure of the WGAN using Gradient-Penalty [23].

The layers of the deep learning model are modified to fit this task based on the dimensions of the data, with 32 nucleotides, each represented by a vector of length 5. The generator consists of a dense layer to increase the dimension of the input noise, following five residual convolution blocks, all using the same number of channels. Finally, the output of the last residual block is fed to a convolution layer with five output channels following a Softmax layer, the output of which is the one-hot encoded sequence (see Table S3). The input is a random vector of length 25, sampled from a normal distribution with a mean of zero and a standard deviation of 1.0. Technically called critic, the discriminator of this model is a convolutional neural network with five residual blocks following a dense layer with one output (see Table S4). Unlike the original DCGAN model [43], there is no activation function at the output of the dense layer of the critic, as the output is an unbounded score and not a probability [3]. The number of iterations of the critic per generator iteration is five here, meaning that the critic is updated five times more than the generator.

As mentioned above, the generative model consists of a generator G and a critic C. The generator gets a random noise as input and generates random samples, and the weights of the network are initialized with a normal distribution range of -0.1 and 0.1. The model's loss function, as shown in Equation 1, is to maximize the distance between the critic's scores for the actual sequences and the generated ones. The first part of the loss function belongs to the original WGAN [3], and the second part includes the gradient penalty and its coefficient λ. Here x and $\tilde{x}$ are the actual and generated samples, where we sample $\hat{x} \sim P_{\tilde{x}}$ uniformly along straight lines between pairs of points sampled from the data distribution $\mathbb{P}r$ and the generator distribution $\mathbb{P}g$ for the computation of the gradient penalty. The gradient penalty improves the training of the GAN by enforcing the Lipschitz constraint as the model proposes [23], and this allows training for a higher number of iterations. Here we train our model for 200,000 epochs.

$$L_{adv} = (\mathbb{E}_{\tilde{x}\sim P_g}[D_{\tilde{x}}] - \mathbb{E}_{x\sim P_r}[D_x]) + \lambda(\mathbb{E}_{\hat{x}\sim P_{\hat{x}}}\left[(||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2\right]) \tag{1}$$

Protein Embedding Model The proposed architecture needs a similarity metric between proteins in order to make use of the similarities between target and related proteins and make inferences about the binding preferences of generated RNA sequences to the targeted protein. In this work, we use the protein embedding model proposed in [4] to extract the $100d$ vector representations of the protein sequences. Using these vector encodings, we calculate the pair-wise Euclidean distances of all proteins, for which a binding assessment model is available, with the targeted protein. The protein embedding model is a deterministic function $f_e(P)$, where P is the sequence information of the RNA. It uses each $3-$mer in the protein amino acid sequence and, using a look-up table (i.e., embedding vectors), it calculates the overall protein embedding by aggregating the embedding vectors for each $3-$mer. We summarize the function of the protein embedding model via the following relation $X = f_e(P)$ where X and P denote the protein embedding vector and corresponding protein amino acid sequence, respectively.

Target Feature Assessment Models In order to guide the generative optimization process, our model needs guidance by which it learns to generate piRNA sequences that have a high binding preference to the targeted protein. To serve this *guidance* purpose, our model uses, what we call, target feature assessment models. Although, in this work, we are using protein binding as our choice of the target feature, it is possible to use any computationally or experimentally assessable target feature to optimize for (e.g., translation rate, half-life).

In this work, we make use of pre-trained DeepBind models. DeepBind models use CNN-based deep networks to evaluate the binding score of any RNA sequence to the protein of interest. DeepBind provides us with a vast array of proteins for which there exists a pre-trained assessment model.

We evaluate the binding score of the generated sequence, at each iteration of the generative optimization, to a set of proteins of interest. We denote the assessment functions with $F_i(R)$, where i enumerates available pre-trained assessment models and R denotes the input RNA sequence. The pre-trained assessment models output target feature scores by which the model guides its generative process. We summarize the assessment relation with $y_i = F_i(R)$, where y_i denotes the assessment score of R, the input RNA sequence, w.r.t. assessing function F_i.

4.5 Training and Optimization

Combiner Block Our model is able to generate piRNA sequences for proteins even when a target feature assessment model is not available for the target protein. The model only needs the amino acid sequence of the protein to be optimized for. Using the corresponding embedding of the target protein, it searches the available assessment models database and finds the proteins which have protein embeddings with minimum Euclidean distance to the target proteins embedding vector. Using a pre-defined number, which we introduce as a hyper-parameter to the model, of the closest proteins (e.g., three of the closest proteins), our model weighs the assessment scores inversely with the corresponding Euclidean distance to the target protein. That is:

$$d_i = \|x_t - x_i\|^2, \tag{2}$$

$$w_i = \frac{e^{1/d_i}}{\sum_{i'=1}^{N} e^{1/d_{i'}}}, \tag{3}$$

$$\hat{y}_t = \sum_{i'}^{N} y_i w_i, \tag{4}$$

where, d_i denotes the distance of the ith closest protein's embedding to the target protein embedding, x_t denotes the target protein embedding, N denotes the number of closest proteins to be used, w_i denotes the

corresponding scalar weight for the ith closest protein and y_i denotes the assessment score associated with the ith closest protein. Hence, the combiner block provides an estimate of the latent function $RB_t(R)$, which we denote by $\hat{y}_t$. Notice how $\hat{y}_t = RB_t(R)$ when $N = 1$ and the target feature assessment model for the target protein is available.

Target Feature Optimization The overall architecture tries to maximize the estimate of the target protein binding score $\hat{y}_t$ by iteratively updating the input latent noise Z with Stochastic Gradient Ascent (SGA). The generation process is described in Algorithm 1.

Algorithm 1: Optimization of Z with the overall architecture

Result: RNA sequences with high binding preference to the targeted protein
$\theta \longleftarrow$ parameters of the trained GAN on any large RNA dataset. ;
$f_e(P) \longleftarrow$ protein embedding function which outputs the embedding vector of its input amino acid sequence. ;
$\{F_i(R)\} \longleftarrow$ set of all available target feature assessment models. ;
$Z = Z_0 \longleftarrow$ batch of initial latent noise sampled from standard normal.;
$P_t \longleftarrow$ amino acid sequence of the target protein.;
$N \longleftarrow$ number of closest proteins to be used.;
$x_t \longleftarrow f_e(P_t)$;
$\{x_i\} \longleftarrow f_e(\{P_i\})$;
$\{d_i\} \longleftarrow \{\|x_t - x_i\|^2\}$;
$\{d_i\}_{i=0}^N \longleftarrow \{d_i\}$ with N of the minimum d_i ;
$\{w_i\}_{i=0}^N \longleftarrow softmax(\{d_i\}_{i=0}^N)$;
while $\hat{y}_t$ *is not converged* **do**
  $R \longleftarrow GAN_\theta(Z)$; `// batch of generated sequences`
  $\{y_i\}_{i=0}^N \longleftarrow \{F_i(R)\}_{i=0}^N$; `// assessment scores for` N `closest proteins`
  $\hat{y}_t \longleftarrow \sum_{i=0}^N y_i w_i$; `// estimate of target protein binding score`
  $\theta \longleftarrow \theta + \nabla_\theta^{\hat{y}_t}$; `// update` θ `with SGA`
end
$R^* \longleftarrow GAN_\theta(Z)$; `// optimized batch of generated sequences`

4.6 *In vitro* experiments

Processes of the production and purification of RNA aptamers and SOX2 protein and rEMSA of binding reaction were described in Figure 6.

Plasmid design and construction The human SOX2 gene (RefSeq ID: NM003106.4) nucleotide sequence with N-terminal $6x$ His-tag was chemically synthesized (Genewiz, NJ, USA). With primers containing suitable overhang sequences homologous to the pET22b(+) cloning vector, the SOX2 gene was cloned into the pet22b(+) vector using Gibson assembly [21]. The reaction was transformed into a chemically competent strain of E. coli DH5α PRO by heat-shock transformation. Selected colonies were verified by NGS Sequencing (Intergen, Turkey). The SOX2 amino acid sequence is listed in Table S1. All of the primers used in this study are listed in Table S2. The pET22b(+) backbone with SOX2 gene map is shown in Figure S2. Sequencing data is shown in Figure S2. Analysis of NGS data was done using Geneious Prime 9.1.8. software.

SOX2 production and purification The construct was transformed into the E. coli BL21 (DE3) strain for expression. Cells from an overnight culture were diluted to 1:100 in LB medium with the proper antibiotics and incubated until the OD_{600} was between 0.4 and 0.6. The culture was then induced with 1 mM IPTG for 24 hours at 16 ℃ and 200 rpm. Using a centrifuge, the cell pellet was collected and resuspended in the lysis buffer (20 mM NaH_2PO_4, 500 mM NaCl, 20 mM imidazole pH 7.4) supplemented with lysozyme (1mg/mL) and PMSF (1mM). First, five cycles of freeze/thaw were applied. Then, for ten cycles of 15s on/45s off, the suspension was sonicated at 30% power. The samples were centrifuged for 1 hour at 21,500 g, and the supernatant was

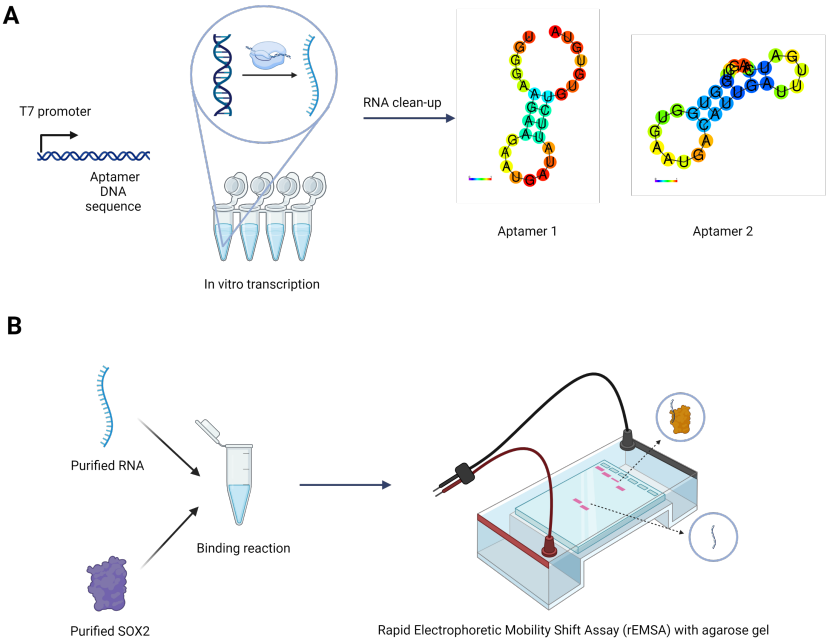

Fig. 6. Experimental pipeline. (A) RNA aptamers were produced using in vitro transcription described by [37]. Produced RNAs were purified and eluted into the binding buffer. The secondary structures of RNA aptamers were obtained from RNAfold WebServer [22]. (B) Purified RNA and purified SOX2 proteins are mixed for binding reaction and run on agarose gel for rapid Electrophoretic Mobility Shift Assay (rEMSA). Bands were visualized with SYBR Safe dye. The image was created with BioRender.com.

then filtered through a 0.45 m filter. Using ÄKTA start protein purification system (GE Life Sciences), the filtered lysate was loaded onto the HisTrap nickel column (GE Life Sciences 17524701) that had already been pre-equilibrated. 10-column volumes of lysis buffer (20 mM imidazole) and 5-column volumes of lysis buffer with an imidazole gradient from 50 to 100 mM were used to wash the column. Finally, five volumes of elution buffer were used to elute proteins (20 mM NaH_2PO_4, 500 mM NaCl, 500 mM imidazole pH 7.4). Using a HiTrap desalting column (GE Life Sciences, 29048684) in ÄKTA start protein purification system, purified SOX2 was desalted into 5 mM Tris–HCl buffer (pH 8.0). Protein concentration is calculated via BCA colorimetric assay with BSA standards (Thermo Fisher Scientific, 23225). Protein production was verified using Western Blotting (Figure S3).

RNA aptamer production and visualization RNA aptamers were produced as described in [37]. Aptamer sequences with T7 promoter sequences at 5'-end *in vitro* were transcribed using HiScribe® T7 High Yield RNA Synthesis Kit (NEB, E2040S), according to the kit's manual. RNA purification was performed using Monarch® RNA Cleanup Kit (NEB, T2040S) according to the kit's manual. Visualization of the RNA aptamers was done using 2x TBE Agarose Gel Electrophoresis (0.13 M tris (pH 7.6), 45 mM boric acid, 2.5 mM EDTA, 0.02 g/mL agarose, 5x SYBR Safe (Thermo Fisher, S33102)) using RNA loading dye (NEB, B0363S) (Figure S4). Gels were visualized using the Fusion Solo S imaging system (Vilber). Gel images are processed using Fiji/ImageJ software.

Aptamer-binding reactions and rapid Electrophoretic Mobility Shift Assay (rEMSA) Binding reactions and gel preparation were performed as described in [42]. Both protein and RNA were suspended in a binding buffer (5 mM Tris–HCl, pH 8.0). Samples were incubated for 20 min on ice and mixed with a 6x loading buffer. High-percentage agarose gel was prepared using 0.5x TB buffer (45 mM Tris base, 45 mM boric acid, pH 8.2) and agarose (0.03 g/ml), 5x SYBR Safe (Thermo Fisher, S33102). The same binding reaction

setup was implemented in testing any nonspecific binding occurring with serum albumin protein (BSA) (SERVA, 11930.01) and observed no shift (Figure S5). Reaction volumes were set to 12 μl. 2.5 μg of protein amounts were used in the binding reactions. Gels were electrophoresed at 200V for 20 minutes. Gels were visualized using the Fusion Solo S imaging system (Vilber). Gel images are processed using Fiji/ImageJ software. Processes of the production and purification of RNA aptamers and SOX2 protein and rEMSA of binding reaction were described in Figure 4. The secondary structures of RNA aptamers were obtained from RNAfold WebServer [22].

Competing Interests: The authors declare no competing interests.

Author Contributions: SB and FO designed and implemented the computational model. DA performed the in vitro experiments. SB, DA, UOSS, and AEC wrote the manuscript. UOSS and AEC supervised the study.

Data Availability: All the data used here are available in public repositories. We downloaded the dataset from the DASHR2-GEO piRNA database [33]. The additional data used in the optimization procedure are available in public DeepBind and Prot2Vec repositories.

Code Availability: The source code, including the implementation of the model, the optimization procedure, and the additional scripts used to generate plots, predict binding scores, and find the closest relative proteins, are released at http://github.com/ciceklab/RNAGEN.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: {TensorFlow}: a system for {Large-Scale} machine learning. In: 12th USENIX symposium on operating systems design and implementation (OSDI 16). pp. 265–283 (2016)
2. Alipanahi, B., Delong, A., Weirauch, M.T., Frey, B.J.: Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. Nature biotechnology **33**(8), 831–838 (2015)
3. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International conference on machine learning. pp. 214–223. PMLR (2017)
4. Asgari, E., Mofrad, M.R.: Continuous distributed representation of biological sequences for deep proteomics and genomics. PloS one **10**(11), e0141287 (2015)
5. Bajan, S., Hutvagner, G.: Rna-based therapeutics: from antisense oligonucleotides to mirnas. Cells **9**(1), 137 (2020)
6. Barazandeh, S., Ozden, F., Hincer, A., Seker, U.O.S., Cicek, A.E.: Learning to generate 5'utr sequences for optimized ribosome load and gene expression. bioRxiv pp. 2023–01 (2023)
7. Bottari, F., Daems, E., de Vries, A.M., Van Wielendaele, P., Trashin, S., Blust, R., Sobott, F., Madder, A., Martins, J.C., De Wael, K.: Do Aptamers Always Bind? The Need for a Multifaceted Analytical Approach When Demonstrating Binding Affinity between Aptamer and Low Molecular Weight Compounds. Journal of the American Chemical Society **142**(46), 19622–19630 (Nov 2020). https://doi.org/10.1021/jacs.0c08691, `https://doi.org/10.1021/jacs.0c08691`, publisher: American Chemical Society
8. Campbell, M.J., Gardner, M.J.: Statistics in medicine: Calculating confidence intervals for some non-parametric analyses. British medical journal (Clinical research ed.) **296**(6634), 1454 (1988)
9. Chopra, A., Balbous, F., Biggar, K.K.: Assessing the in vitro Binding Affinity of Protein–RNA Interactions Using an RNA Pull-down Technique. Bio-protocol **12**(23), e4560 (2022). https://doi.org/10.21769/BioProtoc.4560, `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9729851/`
10. Chuai, G., Ma, H., Yan, J., Chen, M., Hong, N., Xue, D., Zhou, C., Zhu, C., Chen, K., Duan, B., et al.: Deepcrispr: optimized crispr guide rna design by deep learning. Genome biology **19**, 1–18 (2018)
11. Churkin, A., Retwitzer, M.D., Reinharz, V., Ponty, Y., Waldispühl, J., Barash, D.: Design of rnas: comparing programs for inverse rna folding. Briefings in bioinformatics **19**(2), 350–358 (2018)
12. Cliff, N.: Dominance statistics: Ordinal analyses to answer ordinal questions. Psychological bulletin **114**(3), 494 (1993)
13. Dammes, N., Peer, D.: Paving the road for rna therapeutics. Trends in Pharmacological Sciences (2020)
14. Dykstra, P.B., Kaplan, M., Smolke, C.D.: Engineering synthetic RNA devices for cell control. Nature Reviews Genetics **23**(4), 215–228 (Apr 2022). https://doi.org/10.1038/s41576-021-00436-7, `https://www.nature.com/articles/s41576-021-00436-7`, number: 4 Publisher: Nature Publishing Group
15. Esmaili-Taheri, A., Ganjtabesh, M.: Erd: a fast and reliable tool for rna design including constraints. BMC bioinformatics **16**(1), 1–11 (2015)
16. Esmaili-Taheri, A., Ganjtabesh, M., Mohammad-Noori, M.: Evolutionary solution for the rna design problem. Bioinformatics **30**(9), 1250–1258 (2014)
17. Fornace, M.E., Huang, J., Newman, C.T., Porubsky, N.J., Pierce, M.B., Pierce, N.A.: Nupack: analysis and design of nucleic acid structures, devices, and systems (2022)

18. Frogner, C., Zhang, C., Mobahi, H., Araya, M., Poggio, T.A.: Learning with a wasserstein loss. Advances in neural information processing systems **28** (2015)
19. Garcia-Martin, J.A., Clote, P., Dotu, I.: Rnaifold: a constraint programming algorithm for rna inverse folding and molecular design. Journal of bioinformatics and computational biology **11**(02), 1350001 (2013)
20. Gerstberger, S., Hafner, M., Tuschl, T.: A census of human rna-binding proteins. Nature Reviews Genetics **15**(12), 829–845 (2014)
21. Gibson, D.G., Young, L., Chuang, R.Y., Venter, J.C., Hutchison, C.A., Smith, H.O.: Enzymatic assembly of DNA molecules up to several hundred kilobases. Nature Methods **6**(5), 343–345 (May 2009). https://doi.org/10.1038/nmeth.1318, `https://www.nature.com/articles/nmeth.1318`, number: 5 Publisher: Nature Publishing Group
22. Gruber, A.R., Lorenz, R., Bernhart, S.H., Neuböck, R., Hofacker, I.L.: The Vienna RNA Websuite. Nucleic Acids Research **36**(Web Server issue), W70–W74 (Jul 2008). https://doi.org/10.1093/nar/gkn188, `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2447809/`
23. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. Advances in neural information processing systems **30** (2017)
24. Harries, L.W.: Rna biology provides new therapeutic targets for human disease. Frontiers in genetics **10**, 205 (2019)
25. Hellman, L.M., Fried, M.G.: Electrophoretic mobility shift assay (EMSA) for detecting protein–nucleic acid interactions. Nature Protocols **2**(8), 1849–1861 (Aug 2007). https://doi.org/10.1038/nprot.2007.249, `https://www.nature.com/articles/nprot.2007.249`, number: 8 Publisher: Nature Publishing Group
26. Huang, Z., Wang, J., Lu, X., Mohd Zain, A., Yu, G.: scggan: single-cell rna-seq imputation by graph-based generative adversarial network. Briefings in bioinformatics p. bbad040 (2023)
27. Ilia, K., Shakiba, N., Bingham, T., Jones, R.D., Kaminski, M.M., Aravera, E., Bruno, S., Palacios, S., Weiss, R., Collins, J.J., Vecchio, D.D., Schlaeger, T.M.: Synthetic genetic circuits to uncover and enforce the OCT4 trajectories of successful reprogramming of human fibroblasts (Jan 2023). https://doi.org/10.1101/2023.01.25.525529, `https://www.biorxiv.org/content/10.1101/2023.01.25.525529v1`, pages: 2023.01.25.525529 Section: New Results
28. Katsamba, P.S., Park, S., Laird-Offringa, I.A.: Kinetic studies of RNA–protein interactions using surface plasmon resonance. Methods **26**(2), 95–104 (Feb 2002). https://doi.org/10.1016/S1046-2023(02)00012-9, `https://www.sciencedirect.com/science/article/pii/S1046202302000129`
29. Kawasaki, S., Fujita, Y., Nagaike, T., Tomita, K., Saito, H.: Synthetic mRNA devices that detect endogenous proteins and distinguish mammalian cells. Nucleic Acids Research **45**(12), e117 (Jul 2017). https://doi.org/10.1093/nar/gkx298, `https://doi.org/10.1093/nar/gkx298`
30. Kim, J., Quijano, J.F., Kim, J., Yeung, E., Murray, R.M.: Synthetic logic circuits using RNA aptamer against T7 RNA polymerase. Biotechnology Journal **17**(3), 2000449 (2022). https://doi.org/10.1002/biot.202000449, `https://onlinelibrary.wiley.com/doi/abs/10.1002/biot.202000449`, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/biot.202000449
31. Kleinkauf, R., Mann, M., Backofen, R.: antarna: ant colony-based rna sequence design. Bioinformatics **31**(19), 3114–3121 (2015)
32. Krishnan, P., Ghosh, S., Graham, K., Mackey, J.R., Kovalchuk, O., Damaraju, S.: Piwi-interacting rnas and piwi genes as novel prognostic markers for breast cancer. Oncotarget **7**(25), 37944 (2016)
33. Kuksa, P.P., Amlie-Wolf, A., Katanić, Ž., Valladares, O., Wang, L.S., Leung, Y.Y.: Dashr 2.0: integrated database of human small non-coding rna genes and mature products. Bioinformatics **35**(6), 1033–1039 (2019)
34. Levenshtein, V.I., et al.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. vol. 10, pp. 707–710. Soviet Union (1966)
35. Liu, G., Zhang, Y., Zhang, T.: Computational approaches for effective crispr guide rna design and evaluation. Computational and structural biotechnology journal **18**, 35–44 (2020)
36. Lorenz, R., Bernhart, S.H., Höner zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P.F., Hofacker, I.L.: Viennarna package 2.0. Algorithms for molecular biology **6**, 1–14 (2011)
37. Lu, C., Li, P.: Preparation of short RNA by in vitro transcription. Methods in Molecular Biology (Clifton, N.J.) **941**, 59–68 (2012). https://doi.org/10.1007/978-1-62703-113-4$_5$
38. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. The annals of mathematical statistics pp. 50–60 (1947)
39. Merleau, N.S., Smerlak, M.: An evolutionary algorithm for inverse rna folding inspired by lévy flights. bioRxiv pp. 2022–01 (2022)
40. Minuesa, G., Alsina, C., Garcia-Martin, J.A., Oliveros, J.C., Dotu, I.: Moirnaifold: a novel tool for complex in silico rna design. Nucleic acids research **49**(9), 4934–4943 (2021)
41. Nimjee, S.M., White, R.R., Becker, R.C., Sullenger, B.A.: Aptamers as Therapeutics. Annual Review of Pharmacology and Toxicology **57**(1), 61–79 (2017). https://doi.org/10.1146/annurev-pharmtox-010716-104558, `https://doi.org/10.1146/annurev-pharmtox-010716-104558`, _eprint: https://doi.org/10.1146/annurev-pharmtox-010716-104558

42. Perea, W., Greenbaum, N.L.: Label-free horizontal EMSA for analysis of protein-RNA interactions. Analytical Biochemistry **599**, 113736 (Jun 2020). https://doi.org/10.1016/j.ab.2020.113736, https://www.sciencedirect.com/science/article/pii/S0003269720302682
43. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
44. Ramanathan, M., Porter, D.F., Khavari, P.A.: Methods to study rna–protein interactions. Nature methods **16**(3), 225–234 (2019)
45. Ream, J.A., Lewis, L.K., Lewis, K.A.: Rapid agarose gel electrophoretic mobility shift assay for quantitating protein: RNA interactions. Analytical Biochemistry **511**, 36–41 (Oct 2016). https://doi.org/10.1016/j.ab.2016.07.027, https://www.sciencedirect.com/science/article/pii/S0003269716302147
46. Rodgers, G., Austin, C., Anderson, J., Pawlyk, A., Colvis, C., Margolis, R., Baker, J.: Glimmers in illuminating the druggable genome. Nature Reviews Drug Discovery **17**(5), 301–302 (2018)
47. Rubio-Largo, Á., Vanneschi, L., Castelli, M., Vega-Rodríguez, M.A.: Multiobjective metaheuristic to design rna sequences. IEEE Transactions on Evolutionary Computation **23**(1), 156–169 (2018)
48. Santos, R., Ursu, O., Gaulton, A., Bento, A.P., Donadi, R.S., Bologa, C.G., Karlsson, A., Al-Lazikani, B., Hersey, A., Oprea, T.I., et al.: A comprehensive map of molecular drug targets. Nature reviews Drug discovery **16**(1), 19–34 (2017)
49. Seo, M., Lei, L., Egli, M.: Label-Free Electrophoretic Mobility Shift Assay (EMSA) for Measuring Dissociation Constants of Protein-RNA Complexes. Current protocols in nucleic acid chemistry **76**(1), e70 (Mar 2019). https://doi.org/10.1002/cpnc.70, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6391183/
50. Stead, J.A., McDowall, K.J.: Two-dimensional gel electrophoresis for identifying proteins that bind DNA or RNA. Nature Protocols **2**(8), 1839–1848 (Aug 2007). https://doi.org/10.1038/nprot.2007.248, https://www.nature.com/articles/nprot.2007.248, number: 8 Publisher: Nature Publishing Group
51. Tan, C., Gao, Z., Li, S.Z.: Generative tertiary structure-based rna design. arXiv preprint arXiv:2301.10774 (2023)
52. Wayment-Steele, H.K., Kim, D.S., Choe, C.A., Nicol, J.J., Wellington-Oguri, R., Watkins, A.M., Parra Sperberg, R.A., Huang, P.S., Participants, E., Das, R.: Theoretical basis for stabilizing messenger rna through secondary structure design. Nucleic acids research **49**(18), 10604–10617 (2021)
53. Wu, X., Pan, Y., Fang, Y., Zhang, J., Xie, M., Yang, F., Yu, T., Ma, P., Li, W., Shu, Y.: The biogenesis and functions of pirnas in human diseases. Molecular Therapy-Nucleic Acids **21**, 108–120 (2020)
54. Yesselman, J.D., Eiler, D., Carlson, E.D., Gotrik, M.R., d'Aquino, A.E., Ooms, A.N., Kladwang, W., Carlson, P.D., Shi, X., Costantino, D.A., et al.: Computational design of three-dimensional rna structure and function. Nature nanotechnology **14**(9), 866–873 (2019)
55. Zuker, M., Stiegler, P.: Optimal computer folding of large rna sequences using thermodynamics and auxiliary information. Nucleic acids research **9**(1), 133–148 (1981)