UTRGAN: Learning to Generate 5' UTR Sequences for Optimized Translation Efficiency and Gene Expression

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

[1] Department of Computer Engineering, Bilkent University, Ankara, Turkey
[2] Department of Computer Science, Oxford University, Oxford, UK
[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. The 5' untranslated region (5' UTR) of the messenger RNA plays a crucial role in the translatability and stability of the molecule. Thus, it is an important component in the design of synthetic biological circuits for high and stable expression of intermediate proteins. Several UTR sequences are patented and used frequently in laboratories. We present a novel model UTRGAN, a Generative Adversarial Network (GAN)-based model designed to generate 5' UTR sequences coupled with an optimization procedure to ensure a target feature such as high expression for a target gene sequence or high ribosome load and translation efficiency. We rigorously analyze and show that the model can generate sequences that mimic various properties of natural UTR sequences. Then, we show that the optimization procedure yields sequences that are expected to yield (i) 61% higher average expression (up to 5-fold) on a set of target genes, (ii) 53% higher mean ribosome load on average (up to 2-fold for the best 5' UTR), and (iii) a 34-fold increase on average translation efficiency, compared to the initially generated UTR sequences. We also demonstrate that when there is a single target gene of interest, the expected expression increases by at least 37% on average and up to 8-fold for certain genes (up to 32-fold for the best 5' UTR).

Keywords: Generative Models · 5' UTR design · Deep Learning · Gene Expression · Translation Efficiency

1 Introduction

RNA-based therapeutics necessitate both tunability and a long-lasting profile after administration to the body. To achieve this, optimization of both mRNA molecules and carriers is crucial to enhance their stability and promote tissue-specific tropism [46]. The level of protein expression from mRNA therapeutics plays a critical role in various applications such as protein replacement therapies, genome engineering, genetic reprogramming studies, as well as vaccination and cancer immunotherapies [39]. In specific protein replacement therapies, conditions like hemophilia A and cystic fibrosis exemplify the significance of targeted mRNA expression. For instance, Chen et al. demonstrated that to attain the required factor VIII expression in mice using mRNA-laden lipid nanoparticles (LNPs), a concentration of 2 mg mRNA per kilogram of mouse body weight is necessary [12]. Similarly, a study focusing on cystic fibrosis treatment found that a dosage of 0.1 mg/kg/day for two consecutive days was sufficient to restore the function of the CFTR gene in CFTR-knockout mice [38]. A phase 1 study addressing transthyretin amyloidosis utilized a dosage of 0.3 mg/kg to patients, resulting in the expression of Cas9 and the delivery of its single guide RNA to knockout the transthyretin gene, representing a potential treatment for this disease [19]. Conversely, vaccination studies for viruses like Zika [36], Covid [48] [18], and influenza [15], or tumor-associated antigen expression in cancer immunotherapies [40], require a lower dosage of approximately 0.002-0.02 mg/kg injection to induce immunity against these viruses and tumor cells. Thus, designing stable and highly-expressed RNA sequences is desirable [3] [8] [42].

To increase the stability and expression level of the mRNA, optimization of the 5' UTR region is a preferable approach. The 5' untranslated region (5' UTR) of a gene is an important component because it affects the mRNA level and the stability of the produced transcript [28] [43]. Although the 5' UTR of the mRNA is not translated into the protein, it plays a crucial role in regulating the translation process because it contains the ribosome binding site (RBS) through which the ribosome attaches to initiate the translation. For this reason, many UTR sequences have been patented and used in the design of gene circuits [31, 50].

The standard approach for optimizing UTR sequences is by introducing variants in existing target sequences and evaluating the effectiveness using simple algorithmic approaches. Von Niessen et al. construct a library of 5' UTR sequences from the Human genome [50]. Then, they generate synthetic sequences using a genetic algorithm based on sequence characteristics such as GC content, k-mer frequency, and free energy. The generated sequences are then selected with respect to their predicted translation efficiency [31] [10]. Similarly, Von Niessen et al. identify 3' UTR sequences that improve transcript stability [49]. Studies also identify 3' and 5' UTRs for higher gene expression, and the samples are selected from naturally abundant mRNA sequences in Human tissues [7]. These models use machine learning algorithms only to predict sequence features and use those features to evaluate the generated samples. The potential of generative modeling is not fully utilized for this purpose.

Several studies employ Generative Adversarial Networks (GANs) to generate sequences with similar characteristics to the natural biological DNA or RNA sequences. RNAGEN is a framework that enables the generation and optimization of synthetic piRNA sequences with some desired properties, such as binding to target proteins [35]. Similarly, Zirmec et al. propose a framework based on GANs for generating 1,000 bp yeast DNA sequences [52] [54]. They employ a model that uses the regulatory regions to predict the yeast gene expression and utilize this model to optimize the gene expression by updating the entire DNA sequence [53]. However, the model is suitable only for yeast because the DNA sequences are relatively short. Thus, this model generates complete DNA sequences and optimizes the entire sequence, and therefore is not capable of generating the regulatory sequences separately that can be plugged into other target genes of interest. Castillo-Hair et al. only theoretically discuss the possibility of using machine learning methods along with predictive models for optimizing 5' UTR sequences [11]. To the best of our knowledge, there is no prior work that is tailored for generating and optimizing regulatory regions, including 5' UTRs.

In this work, we propose UTRGAN, the first GAN-based pipeline for generating novel Human 5' UTR sequences and optimizing them for higher predicted Mean Ribosome Load (MRL), Translation Efficiency (TE), and higher gene expression. Our model generates 5' UTR sequences that can be attached to any real or synthetic gene of interest. UTRGAN can generate 5' UTR sequences without modifying the target DNA/RNA sequence it is attached to. Our generated sequences resemble the natural 5' UTRs with respect to several important characteristics, such as GC content distribution, k-mer distance, and Minimum Free Energy (MFE) [45]. Furthermore, the optimization procedure enables the generation of optimized sequences for higher MRL, TE, and mRNA abundance. By using multiple sequences for optimization, we increase MRL and mRNA expression by 53% and by 61% compared to the initial values, respectively. In addition, optimization for TE results in up to a 34-fold increase in the average predicted value. Depending on the application, our model is able to optimize the generated 5' UTRs on a specific target DNA sequence or on a set of DNA sequences. The optimization for a single gene of interest increases the expression 2.2-fold on average and can result in up to 32 times higher expression for the best synthetic 5' UTR. UTRGAN's ability to generate and optimize 5' UTRs for any given gene sequence will be the key enabler for the genetic circuit design. We think that UTRGAN will pave the way for mRNA-based therapeutics in the biotech industry for any application that requires gene expression control, such as cancer immunotherapy.

2 Results

2.1 Overview of UTRGAN

The UTRGAN model is a deep generative adversarial network (GAN) that learns generating 5' UTR sequences with similar characteristics to the natural ones. This model is a novel version of the WGAN-GP architecture [21] for one-hot encoded DNA sequence. The model provides improved training over the original Convolutional GAN (DCGAN) models [21] [37] and is less prone to overfitting than the original WGAN architecture [5].

Figure 1A shows the overview of the 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 5' UTR sequences, whereas the critic uses convolutions and dense layers to score the natural and generated samples. The optimization pipeline is shown in Figure 1B where off-the-shelf deep-learning models are used for optimizing the gene expression [2] given an RNA, and MRL [23] or translation efficiency [51] given a 5' UTR sequence. We perform optimization by updating the initial input vector until all

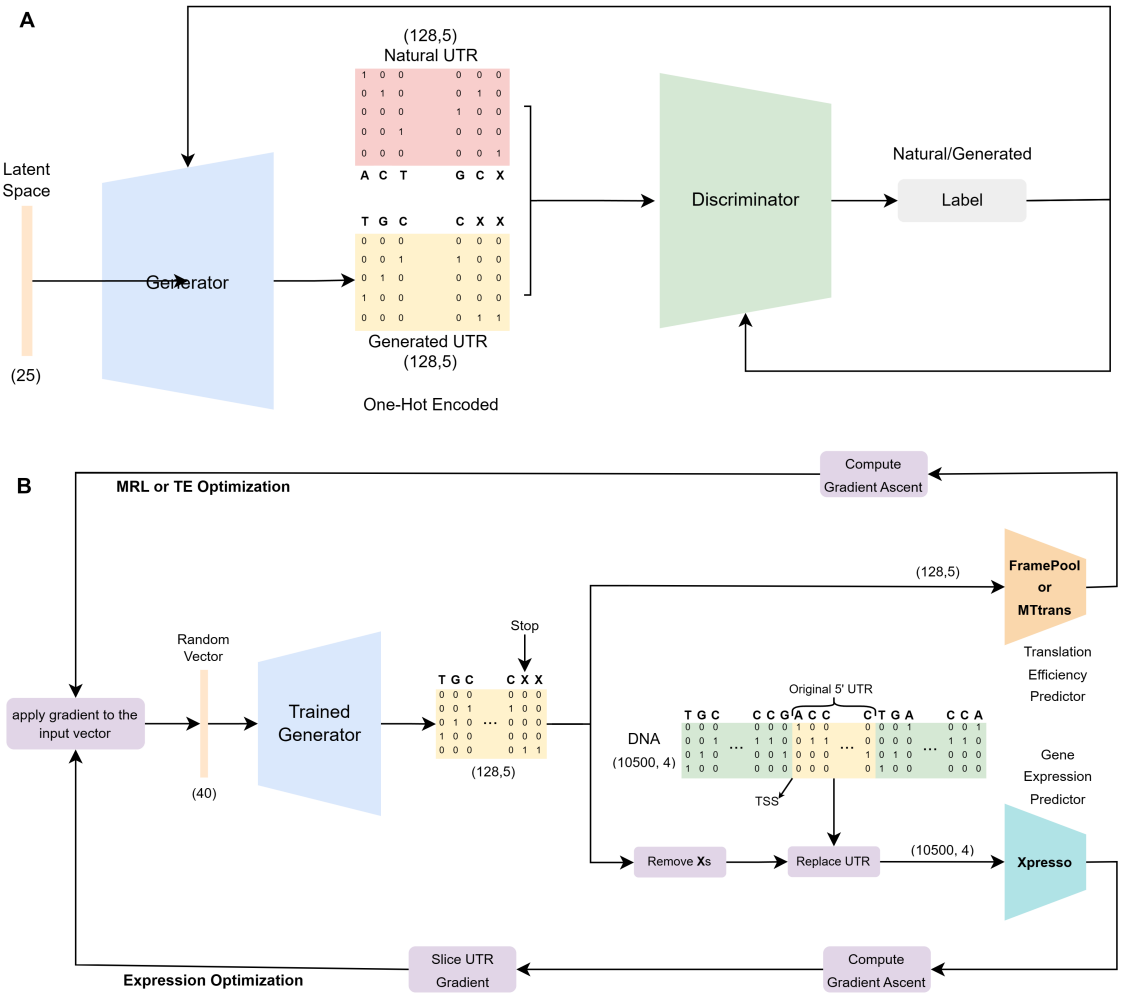

Fig. 1: **Generative and optimization architectures.** Shown on **A** is the training phase of the GAN model, where the competition between the Critic and the Generator and updating their weights based on the types of samples is used to train the models. The diagram of the pipeline shown on **B** is where the optimization takes place. The GAN is used to generate samples, and the input noise is updated using the Gradient Ascent algorithm to increase the predicted MRL, TE, or mRNA abundance of the generated 5' UTR sequences. These two paths with different considerations are for translation efficiency (MRL and TE) and mRNA optimization. The mRNA optimization requires attaching the 5' UTR to a DNA and then slicing the relevant gradient to update the input noise, whereas the MRL and TE get 5' UTRs as their input. Therefore, the optimization is more straightforward for MRL and TE compared to mRNA abundance.

the UTRs converge to their maximum predicted values. That is, we apply Gradient Ascent on the generator's input, as the target is to maximize the MRL, translation efficiency, or gene expression values.

2.2 Levenshtein distances are similar in natural and generated UTRs

A metric to measure the similarity among two sets of sequences is to use the distribution of the distances to the closest sequences in the target (i.e., natural) set. Levenshtein distance is the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one word into the other [27]. To measure the distance between the set of generated and random sequences in relation to the natural dataset, we find the distance of each generated and each random sequence to the closest sequence in the natural samples in terms of Levenshtein distance. This gives us two distributions: One for the generated and one for the random set. To also compare with the natural UTRs, we perform a similar analysis on the natural sequences. That is, we find the smallest non-zero distance for each natural sample (i.e., the nearest neighbor) in the natural UTR set. The distributions shown in Figure 2A demonstrate that the generated sequences have a nearly identical distribution of distances to the distribution within the natural sequences, whereas randomly generated samples display a distinct distribution. Furthermore, the average distance of the random sequences to the closest natural sequence is $\sim$80, while this number is $\sim$50 for generated and $\sim$55 for natural sequences. This indicates that the generative model effectively learns the contextual characteristics of the UTR sequences while it generates novel sequences with a certain minimum distance to the natural samples. There are a handful of natural sequences with anomalously lower distances to the natural UTR set, and we discard those in the plots. We used the two-sided Mann-Whitney U-test as a non-parametric statistical test in order to compare different pairs of distributions [32] shown in Figure 2. We also used the non-parametric delta statistics (Cliff's delta) to calculate the effect size in our experiments [13] and the 95% confidence interval of the medians for the sets of values are also calculated [9]. We used 2048 random, 2048 generated, and all the 33,250 natural sequences in this analysis. The same number of sequences is used for the experiments in the following subsections unless stated otherwise. The p-value for the mentioned test between the set of natural and generated sequences was 0.34, with an effect size of 0.01 and a 95% confidence interval of [0, 0]. The same test resulted in a p-value of 0 between the random and natural sequences (effect size: 0.84, 95% confidence interval: [8, 9]).

2.3 The generative model maintains the 4-mer distribution of the sequences

Another metric to measure the similarity among two sets of sequences is to use the distribution of the k-mers within the sequences. We compute the frequency of the 4-mers for each sequence as also done in the literature [47]. Then, for each sequence in the generated, natural, and random set, we find the closest sequence (Euclidean distance) in the entire set of natural 5' UTRs, ignoring identical sequences (so natural sequences match the nearest neighbor). We obtain a distribution per group. We observe very similar distance distributions for the generated and natural samples as shown in Figure 2B, indicating that the generated set of sequences retain structural similarity to natural sequences and yet, are distinct in terms of the sequence, as also shown in the previous subsection (Mann-Whitney U-test, p: 0.51, effect size: -0.02, 95% confidence interval: [-0.18, 0.09]). In accordance, we observe that the random sequences have a different 4-mer distribution despite being similar to natural sequences in terms of the sequence itself (Mann-Whitney U-test, $p<0.01$, effect size: 0.89, 95% confidence interval: [2.53, 2.92]). Similar to the previous subsection, we discard a few of the natural sequences due to their anomalously lower distances to the set of natural sequences.

2.4 GC content distribution is consistent in the generated 5' UTRs

The percentage of guanine-cytosine content (GC content) in a sequence is important and informative on the stability molecule [24]. We expect the distribution of the GC content in the natural and generated sequences to be similar. We observe that the mean GC content in the generated, random, and natural samples are very similar, and the generated sequences cover a wide range of GC content (Figure 2C), and the Mann-Whitney U-test resulted in a p-value larger than 0.01 (effect size: -0.03, 95% confidence interval: [-0.013, 0.001]). This is in contrast to the GC content for random sequences that are limited to a smaller range: Between 44 and 75 (Mann-Whitney U-test, p: 0.15, effect size: 0.01, 95% confidence interval: [2.53, 2.92]). The higher p-value and the small effect size for random sequences are expected as we generated the random sequences using the average GC content of the set of natural sequences, and despite the smaller range, the averages are aligned, resulting in a larger p-value and smaller effect size.

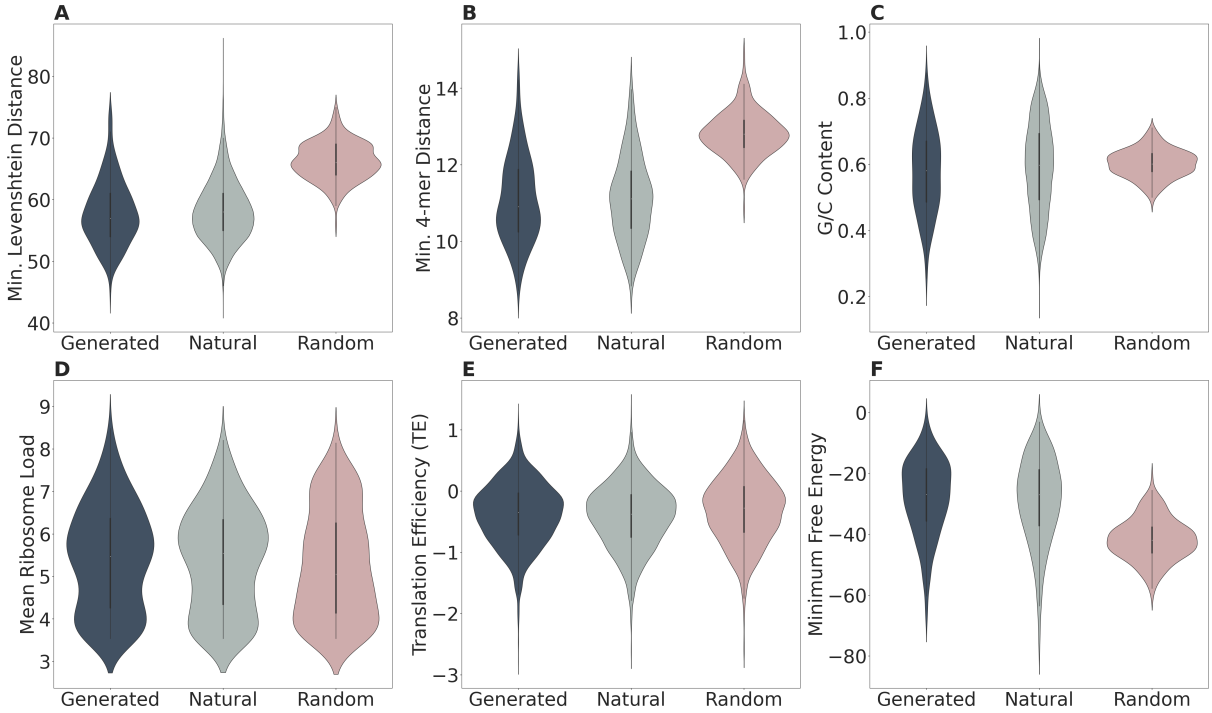

Fig. 2: **Comparison of generated, natural, and random sequences. A**) We calculate the Levenshtein distance to the closest non-identical natural sample; The distribution of these values in generated sequences is more similar to the natural sequences, and the average minimum distance of the random sequences is higher, showing that the random ones have a different minimum distance distribution (random samples p-value: 0.0, generated samples p-value: 0.34). **B**) We calculate the 4-mer frequencies in each sequence and select the distance to the closest non-identical natural sample; The distribution of these values in generated sequences is more similar to the natural sequences, and the average minimum distance of the random sequences is higher, showing that the random ones have different 4-mer frequency distributions (random samples p-value$<$0.01, generated samples p-value: 0.51). **C**) The random sequences are generated based on the average GC content of the set of natural 5' UTRs. Although the average GC content is controlled, the distribution of the GC content is not similar in natural and random sequences. On the other hand, this distribution for the generated sequences is aligned with the natural ones (random samples p-value: 0.15, generated samples p-value: 0.01). **D**) The distributions of MRL for the generated and natural sequences are almost identical. Even though the random sequences also cover the same range, they are distributed differently. Random sequences have variable lengths resembling the length of generated/natural sequences (Mann-Whitney U test, random samples p-value$<$0.01, generated samples p-value: 0.91). **E**) Similar to MRL, we use the MTtrans model to predict the translation efficiency of a batch of generated sequences, and as shown here, the generated ones are similar to natural sequences, while the distribution of predicted *Log*-TE in random sequences is not exactly aligned to that of natural samples (random samples p-value$<$0.01, generated samples p-value: 0.92). **F**) The distribution of MFE for the generated sequences resembles that of natural sequences. The random sequences are distributed considerably differently and cover a smaller range of MFE values (random samples p-value$<$0.01, generated samples p-value: 0.17).

2.5 Predicted Mean Ribosome Load and translation efficiency of the natural and generated have similar distributions

While the above-mentioned metrics are informative about the performance of the model in generating samples structurally similar to the natural sequences, they do not reflect information on the function. MRL is a metric defined based on the ribosome count associated with an mRNA molecule and is considered a proxy for translation rate [23]. We use MRL to measure the similarity of the sequences from a functional perspective.

To estimate the MRL of a given 5' UTR sequence, we use a convolutional neural network model (FramePool) that predicts the MRL of a 5' UTR sequence [23]. This model utilizes a Frame-Pooling layer which processes the frames of three consecutive nucleotides (i.e., codons). Figure 2D shows that the MRL distributions of the natural and the generated sequences are almost identical (p: 0.91, effect size: 0.003, 95% confidence interval: [-0.04, 0.04]). In contrast, the random sequences are more likely to generate lower MRL values (p<0.01, effect size: -0.06, 95% confidence interval: [-0.19, -0.07]).

Similarly, we use the MTtrans model trained on ribosome profiling datasets to predict the translation efficiency of the UTR sequences [51]. Translation efficiency is another proxy for the rate that a cell translates mRNA. MTtrans is trained on various translation profiling datasets, including (i) 3 massively parallel assay (MPRA) polysome profiling datasets and (ii) 3 ribosome profiling datasets. The model trained on MPRA datasets is called *3M*, the model trained on ribosome profiling datasets is called *3R*, and the model trained on all datasets is called *3M3R*. We use the MTtrans 3R version. As shown in Figure 2E, the predicted TE for generated sequences is in the same range as the natural sequences (p: 0.92, effect size: 0.001, 95% confidence interval: [-0.02, 0.02]). Although the random sequences have a slightly higher predicted *Log*-TE, they are not distributed similarly compared to the natural sequences (p<0.01, effect size: 0.10, 95% confidence interval: [0.07, 0.11]).

2.6 Generated and natural samples have similar Minimum Free Energy distributions

The MFE of a 5' UTR sequence as part of the RNA sequence is another informative characteristic of the sequence regarding its stability. It depends on the number, composition, and arrangement of the nucleotides in the mRNA sequence [45] and the 5' UTR in our case. Similar to the GC content, even though MFE might not be the sole indicator of any characteristic of a given 5' UTR sequence, we expect to see similar distributions of MFE values in a large number of samples. As shown in Figure 2F, the distribution of this value is very similar in generated and natural sequences (p: 0.17, effect size: 0.01, 95% confidence interval: [-0.19, 1.0]) while very different in the set of randomly generated sequences (p<0.01, effect size: -0.59, 95% confidence interval: [-14.10, -13.19]). We calculate the MFE of the sequences we use two packages Nupack [16] and ViennaRNA [30] [22]. The result with ViennaRNA package is presented as the results are similar.

2.7 Optimization yields sequences with higher expected expression

The optimization procedure, as explained in Section 4, is based on an iterative procedure that updates the input noise of the GAN to generate 5' UTR sequences with higher expression. We use the model to fine-tune the sequences for higher predicted expression. The model used for prediction is a convolutional neural network-based model (Xpresso [2]) that outputs the predicted log TPM expression for the input DNA sequence. The mean saliency scores in the Xpresso paper [2] is used used to measure which nucleotides around the TSS affect the predicted expression more. The authors show that the most important ones are downstream of the TSS, where the 5' UTR is placed. Based on this analysis of the Xpresso model, we expect the optimization of the 5' UTR part of the DNA alone to increase the expression value.

As shown in Figure 3A, the optimization of 100 generated 5' UTR sequences for maximizing the average expression of a set of 8 randomly selected Human genes (*MYOC*, *TIGD4*, *ATP6V1B2*, *TAGLN*, *COX7A2L*, *IFNGR2*, *TNFRSF21*, *SETD6*) results in increased average predicted mRNA expression in 80% of the DNA samples with the optimized 5' UTRs. To calculate a single score for each 5' UTR, we average the expression over all 8 genes with the generated 5' UTRs replacing the original ones. Note that we are limited by updating the 5' UTR only, and it is not the sole determinant of the expression. Yet, we observe up to 61% increase in the average predicted gene expression for compared to the initial value. In 20% of the cases, the optimization results in reduced expression compared to the initial generated sequence. In those cases, the user can pick the initially generated 5' UTR. We discuss the possible reasons for degradation in the Discussion Section. In

addition, our optimized 5' UTR resulted in a 38% increase in the average predicted expression of a distinct set of 8 randomly selected genes (*ANTXR2*, *NFIL3*, *UNC13D*, *DHRS2*, *RPS13*, *HBD*, *METAP1D*, *NCALD*). This indicates that the optimization is capable of generalizing 5' UTRs for higher average gene expression.

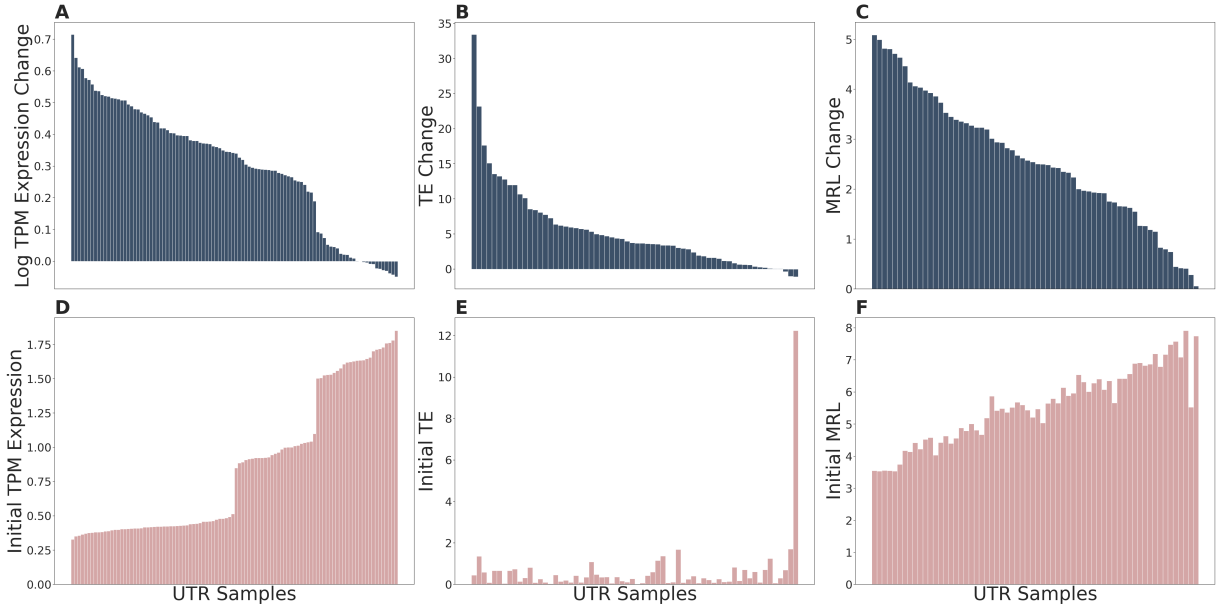

Fig. 3: **Overall performance of gene expression, TE, and MRL optimization A)** The expression change after 3,000 iterations of optimization for 8 DNA samples shows that the model successfully generates 5' UTR sequences and optimizes a majority of those to improve the original expression. The values are log TPM Expression values, and the optimization yields higher predicted gene expression in 80% of the DNAs. **B)** Here is shown the translation efficiency change of 64 optimized generated 5' UTR sequences. In this case, a 97% majority of the sequences yield substantially higher TE after optimization. **C)** Similar to gene expression and TE, optimization is effective for the majority of the sequences and performs better for sequences with low initial MRL values. **D, E, and F)** Here, we show the initial gene expression, TE, and MRL values for the corresponding samples in panels D and E, and F, respectively. For all three panels, initial values tend to increase towards the right. We see that it is more likely for the optimization to degrade performance for samples with high initial TE or gene expression values. Unlike gene expression and TE, degradation in the predicted MRL of the optimized 5' UTRs is rare.

2.8 Mean Ribosome Load and Translation Efficiency increases after optimization

Unlike the mRNA abundance, the MRL is not a DNA-specific metric and does not require a target DNA or mRNA sequence for optimization. The FramePool [23] model can predict the MRL value for 5' UTR sequences of any length and does not require the UTR sequence to be attached to an mRNA sequence. We use this model and optimize the initially generated 100 5' UTRs for 10,000 iterations. The results show increased MRL in more than 95% of the generated 5' UTRs (See Figure 4A). The number of used sequences can be increased, but we would expect similar results as each sequence is optimized separately, and the gradients are not merged.

Another model we use to optimize our generated 5' UTRs is the MTtrans model [51], which predicts translation efficiency. Translation efficiency is another proxy for the rate that a cell translates mRNA. MTtrans is trained on various translation profiling datasets, including (i) 3 massively parallel assay (MPRA) polysome profiling datasets and (ii) 3 ribosome profiling datasets. The model trained on MPRA datasets is called *3M*, the model trained on ribosome profiling datasets is called *3R*, and the model trained on all datasets is called *3M3R*. Here, we use the MTtrans *3R* model to optimize generated 5' UTRs for higher translation efficiency (*Log*-TE). The results in Figure 4B show that the optimization yields a substantial increase in *Log*-TE values. To combine

the optimization for mRNA abundance and translation efficiency, one can optimize many sequences for gene expression first and then select the ones also with higher MRL or translation efficiency. We also present our two-step optimization results in Section 2.10.

In addition to the TE prediction using the 3R model, the MTtrans 3M model allows us to extract important motifs from the MPRA datasets used for training the model [41]. It outputs a set of 256 7-mers as motifs that affect the predicted MRL more than others, both positively and negatively. During MRL optimization using the FramePool model, we observed that the number of motifs affecting the MRL negatively was reduced by 53% on average after optimization, while positive motifs were mostly preserved or increased. This shows that optimization using our model eliminates the negative motifs based on the gradients backpropagated from the MRL prediction model.

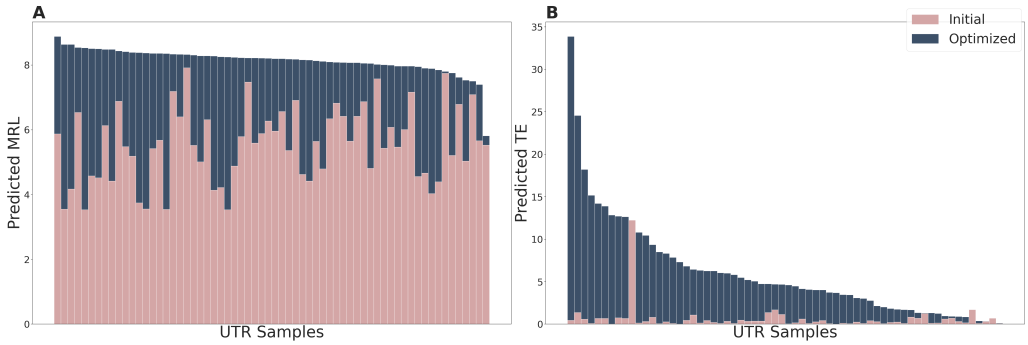

Fig. 4: **Overall performance of MRL and TE optimization. A)**. The optimization for higher MRL using FramePool results in sequences with a considerably high average MRL of 7.6. The optimization result may vary depending on the initial random sequences, and it optimizes almost all sequences to MRLs higher than 7. **B)**. Translation efficiency reaches a high value of 33 after optimization. The initial values are all around the average *Log*-TE of the natural samples, and optimization increases the average more than 32-fold, and the highest optimized value is close to the maximum value of the natural sequences.

2.9 Optimization improves the expected expression for specific target genes

As discussed above, our optimization procedure increases the gene expression when we optimize multiple generated 5' UTRs for multiple gene sequences (optimization based on average). Instead, we can work with a single target gene as well. To optimize 5' UTRs for a specific gene, we generate 128 sequences using a fixed seed and optimize them for higher expressions when placing the UTR of the target gene only.

We optimize 5' UTRs for the *TLR6*, *IFNG*, *TNF*, and the *TP53* genes from the Human genome. Noreen and Arshad [33] discuss the role of the *TLR6* gene in the regulation of innate as well as adaptive immunity. Based on its role in the immune system, the expression level of this gene can affect the immune response. *IFNG* (*IFN*-γ) gene also has a role in the immune system and encodes an important cytokine, interferon-gamma, for immune response. [6, 44]. Similarly, *TNF* or *TNF*-α is the tumor necrosis factor gene and encodes cytokines that play a crucial role in the immune system's inflammatory response. Finally, *TP53* gene is one of the most important tumor suppressor genes and encodes the P53 protein [26, 34]. Controlling the expression of these genes has important implications in the immunology and oncology fields. These are provided as examples to show the capacity of the model to increase the expression of specific target genes. This approach can be used on any gene for expression optimization, both to increase and decrease expression.

We show in Figure 6 that optimization for these genes improves gene expression for more than 95% of the generated 5' UTRs. It provides a 4-fold, 37%, 63%, and 4.2-fold increase in expression on average for each gene, respectively. These results show that optimization is successful on average, and users can pick the top generated sequence for their application. The performance increase for the highest expression yielding 5' UTR is 8-fold, 3-fold, 32-fold, and 3.3-fold for each respective gene. The sequences of the best-performing 5' UTRs for these genes are provided in Supplementary Table 1.

Moreover, we observe that the GC content of the sequences increases with gene expression optimization, and the best-performing sequences exhibit GC content as high as 85%. While the GC content of some natural sequences also exceeds this number, it is important to acknowledge that although high GC content can increase mRNA expression levels [4] [25], it may not be desirable in all circumstances [24]. To address this concern, we have incorporated a control mechanism in our optimization process, allowing us to limit the maximum level of GC content for the selected best-performing sequences. For details on the optimization results with an upper bound of 65% for the GC content of the optimized sequences, please refer to Supplementary Figure 1.

2.10 Joint optimization results in both higher translation efficiency and gene expression

As shown in Sections 2.7 and 2.8, independent optimizations for translation and mRNA abundance yield positive results. Nevertheless, optimizing one does not necessarily increase the other. Here, we also consider the case where optimizing 5' UTR sequences for both higher translation rate and mRNA abundance for a target gene. We use the same set of 4 genes used in Section 2.9. We use a sequential optimization procedure to achieve this goal. Since TE optimization is gene agnostic and mRNA abundance optimization is gene-specific, we first optimize the input vector z for higher translation efficiency for 1,000 iterations. Then, we use the optimized z as the starting point for mRNA abundance optimization. The optimization first increases the average *Log*-TE from negative to positive. That is, the average *Log* value is increased from -0.56 to 1.49. Then, the TE-optimized sequence is further optimized for mRNA abundance for each of the target genes for 1,000 iterations. Although the second optimization procedure slightly decreases the average translation efficiency from 1.49 to slightly above zero for certain genes, both the translation efficiency and mRNA abundance are higher compared to the initially generated sequences, as shown in Figure 5. In addition, we observe that our two-step optimization performs better than optimizing for these two scores simultaneously. The sequences of the best-performing 5' UTRs after joint optimization for these genes are provided in Supplementary Table 2.

2.11 Synthtetic 5' UTRs improve the predicted expression of target genes with natural 5' UTRs

Our main goal in this study is not to replace existing 5' UTRs of natural genes. The experiments in the previous subsections with a single or a set of target genes are to show that the optimization mechanism can tailor the initially generated 5' UTR sequences toward a goal (maximize expression) using natural gene sequences as templates/examples when their UTR sequences are replaced with the generated and the optimized UTR sequences.

Our approach is rather intended for novel and synthetic DNA molecules with some desired properties, such as binding to another protein after translation. Yet, we investigate if the 5' UTRs we generate yield higher predicted expression compared to the natural 5' UTRs. For each of the *TLR6*, *IFNG*, *TNF*, and *TP53* genes, we generate 128 5' UTR sequences and pick the sequence that maximizes the expected expression and compare this value with the expected expression of the gene using their natural 5' UTRs. We observe that the sequences we generate yield 8-fold, 3-fold, 32-fold, and 3.3-fold increases in the expected expression, respectively.

3 Discussion

We introduce UTRGAN, the first framework for developing synthetic 5' UTR sequences optimized towards a target feature. Our approach enables us to generate sequences with similar attributes to the Human 5' UTR samples without the need for exploring intractable search space of DNA sequences that have lengths of up to 128 nucleotides. We advance the state-of-the-art, which relies on optimizing existing UTRs via modifications and by generating sequences from scratch. Our model can also generate sequences of variable lengths, further increasing the possible space of generated sequences.

UTRGAN is a robust framework for synthetic 5' UTR 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' UTR dataset and state-of-the-art prediction models for MRL, TE, and mRNA abundance [23] [51] [2]. Our model can potentially optimize sequences for given any objective function as long as a differentiable prediction model exists to predict the target metric for a 5' UTR sequence. We can maximize and minimize the target feature. 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.

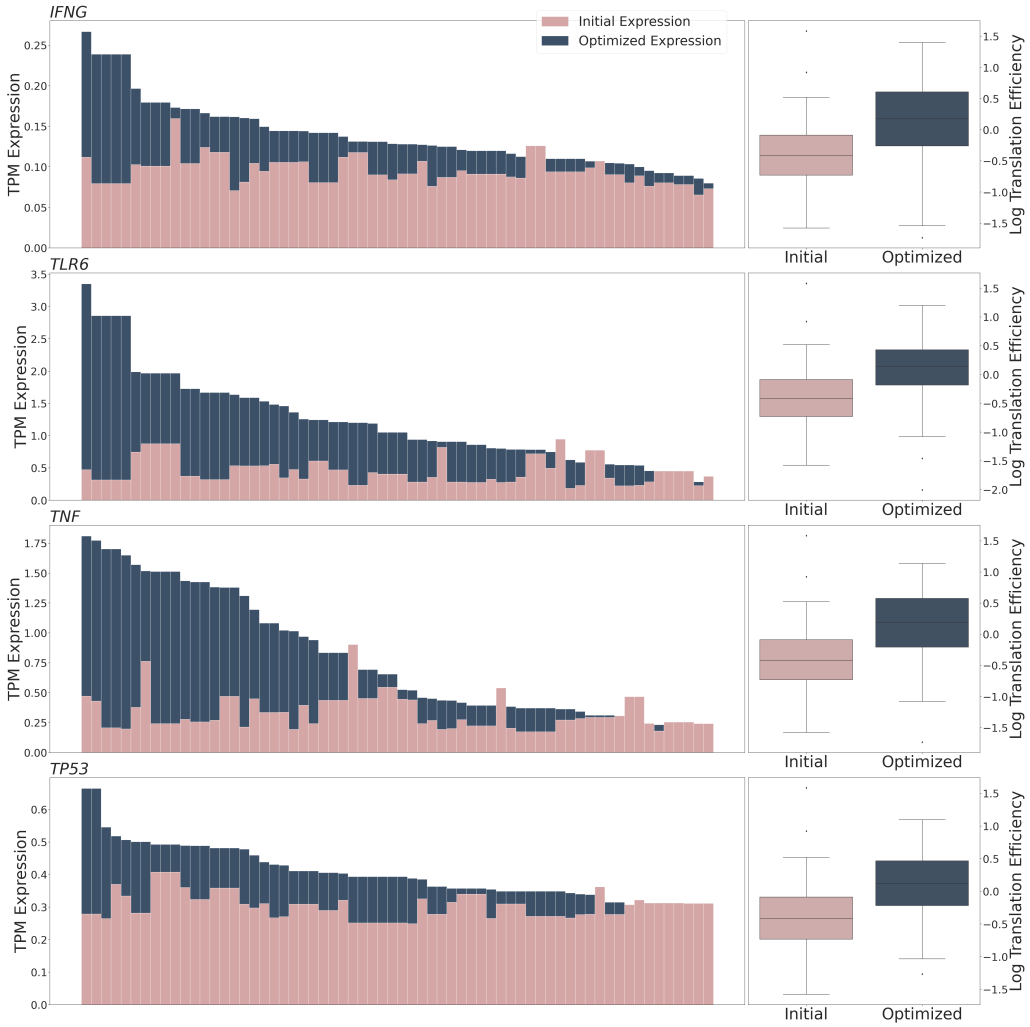

Fig. 5: **Jointly optimizing for TE and mRNA expression** The generated sequences are first optimized for high TE for 1,000 iterations, and then the input vector resulting from the optimization is used as the starting latent vector for gene expression optimization for another 1,000 iterations. As seen for the four genes, *TLR6*, *IFNG*, *TNF*, and *TP53*, the two-step optimization results in higher gene expression and translation efficiency. The predicted gene expression increases up to 12-fold, 1.8-fold, 10-fold, and 2.8-fold with respect to the natural 5' UTRs of the genes, respectively. The maximum predicted TE after optimization is at least 77% higher than the natural 5' UTR of the target genes. The box plot characterizes the TE values using the 25th, 50th, and 75th, also known as quartiles (Q1, median, Q3) and the interquartile range (IQR = Q3 - Q1), with whiskers extending to a maximum of 1.5 times the IQR. Outliers beyond the whiskers are plotted separately.

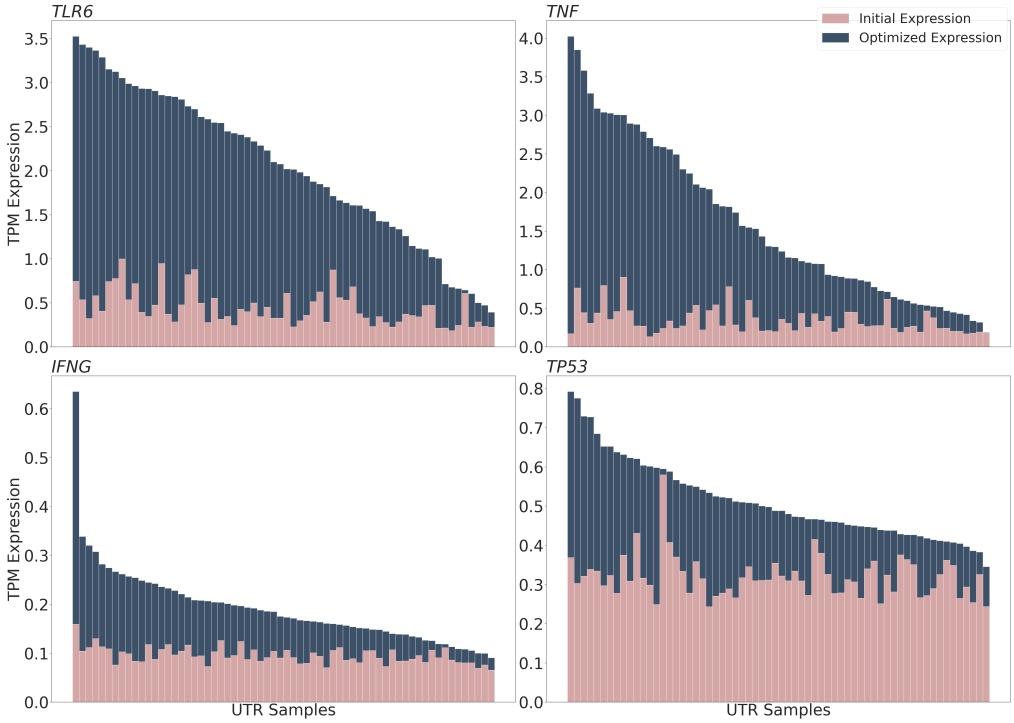

Fig. 6: **Expression optimization for specific target genes.** In addition to optimizing for random genes, it is possible to optimize many generated UTR sequences for specific genes. Here is shown the result of optimization for four specific genes. *TLR6*, *IFNG*, *TNF*, and *TP53* genes show a substantial increase in the expression for optimized sequences compared to initial predicted expression values. The optimization shows 4-fold, 37%, 63%, and 4.2-fold increase in expression on average for *TLR6*, *INFG*, *TNF*, and *TP53*, respectively.

Although UTRGAN can generate and optimize sequences with desirable characteristics, the optimization can lead to decreased TE or mRNA in less than 5% of the cases for TE in the worst optimization and 20% of the cases for mRNA abundance for optimizing UTRs for multiple genes, while it rarely happens for MRL optimization (Figure 3C). One possible reason for decreased expression is the limitation with regard to the predictor model (Xpresso). It operates on the entire DNA (gene) sequence, and while back-propagating the gradient to update the sequence, we can only use part of the gradient that corresponds to the 5' UTR. Thus, changing the 5' UTR but not the rest of the sequence might lead the model to end up at a worse point in the loss space, and changing the rest of the DNA is not desirable for the target application, which requires the rest of the DNA to remain unchanged. Furthermore, in both mRNA expression and TE optimization, we observe that the decreased scores tend to occur when the optimization procedure starts with a sequence that yields a high score, as seen in Figure 3(Panels D and E). In such a case, users can use the starting sequence and discard the optimization.

We expect to see wide use of data-driven sequences generated using deep learning methods in the near future. Generating 5' UTRs utilizing this approach is practical and does not require massive computational resources. While we focus on 5' UTRs, it is straightforward to generalize the framework to generate 3' UTRs or other regulatory elements. It is also possible to use other generative models, such as variational autoencoders or diffusion models. Generating sequences with regulatory roles enables researchers to conduct experiments in a much shorter time, which can be particularly beneficial in mRNA vaccine production and mRNA-based therapeutics.

4 Methods

4.1 Datasets and the Experimental Setup

We obtain the natural UTR sequence data from UTRdb 2.0 [29], a complete database of 3' and 5' UTR sequences of various organisms. We use only the Human 5' UTR samples. The length of the sequences ranges from 30 bp to 600 bp. However, this range is too wide for a CNN model since all the sequences should be padded up to the longest sequence, making padding longer than the sequence itself in most cases. Therefore, we limit the length of the training between 64 bp and 128 bp. This subset contains 33,250 natural 5' UTR sequences. We use 90% of the samples for training and 10% of the samples for validation. We train the model with batches of size 64. We obtain the DNA sequences of Human genes used for optimization from the Ensembl Biomart tool [14].

We encode the data using one-hot encoding, and the dimension of each UTR is $(128, 5)$, where 128 is the maximum size. and 5 is the dimension of the one-hot vector for each nucleotide. The DNA 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), T:(0,0,0,1,0), X:(0,0,0,0,1) where X is the padding character used when the sequence is shorter than 128 bp. This padding is necessary because the input of the 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 64 and 128 nucleotides and then select each nucleotide randomly from the DNA alphabet.

The model is implemented using the Tensorflow 2 library [1] 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 14 hours for 3,000 epochs, and we select this checkpoint based on the validation loss as well as the p-values obtained from the statistical tests as shown in Figure 2. The TE, MRL, and gene expression optimization take 3, 4, and 15 minutes, respectively. Optimizing UTR sequences for higher average expression in 8 genes takes 90 minutes. The code is available at https://github.com/ciceklab/UTRGAN.

4.2 The Model

Overall Architecture
The generative model used here is a variant of a Generative Adversarial Network (GAN) [20]. The input of the GAN is a random noise sample z, and we train the GAN to generate realistic 5' UTR sequences. The GAN learns to map the noise to the 5' UTR space. Then, we can optimize the model to generate specific sequences by optimizing the input noise only. The optimization performed here does not update the weights of the generator and changes the input noise z instead.

The sequences generated by the GAN are scored by the selected scoring models. We use three deep convolutional neural networks to score the generated sequences. The mRNA abundance scoring model [2] predicts the log TPM expression of a given gene sequence, including a 5' UTR. Xpresso provides three models for predicting median expression among many cell types, K562 erythroleukemia cells and GM12878 lymphoblastoid cells [2]. We use the median prediction model in this study, but the optimization can be performed for specific cell types using the mentioned models. The models used to predict the ribosome load and translation efficiency of the 5' UTR requires only the 5' UTR sequence as input [23] [51]. We use the MTtrans 3R model that predicts *log*-TE values as a proxy for translation rate by inputting only the 5' UTR sequence [51]. All models and the generator are differentiable, which enables us to optimize the input z with Stochastic Gradient Ascent by updating it in multiple iterations. Sequences are fine-tuned for 3,000 iterations for optimizing expression and 10,000 for optimizing MRL and TE. iterations and the performance in each iteration is stored. The version with the top performance is picked as the final optimized version of that sequence.

Generative Model Architecture The GAN used here is based on the original Wasserstein GAN (WGAN) [5] model that is an improved version of the Deep Convolutional GAN (DCGAN) model [37] with the Wasserstein loss [17]. 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 [21].

The layers of the deep learning model are modified to fit this task based on the dimensions of the data, with 128 nucleotides, each represented by a vector of length 5. The generator G 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 Supplementary Table 6 for layer dimension details of the generator. The input is a random vector of length 40, sampled from a normal distribution with a mean of zero and a standard deviation of 1.0. Technically called critic, the Discriminator D is a convolutional neural network with five residual blocks following a dense layer with one output. See Supplementary Table 7 for details of the critic layers. Unlike the original DCGAN model [37], 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 [5]. The parameters of the critic are updated 5 times for every update performed on the generator. This means that the critic is updated 5 times more than the generator.

The initial weights of the network are sampled from a normal distribution with mean = 0 and standard deviation = 0.1. In the loss function, as shown in Equation 1, the generator learns to minimize the distance between the distribution of the critic's scores for the natural and generated sequences. The first part of the loss function belongs to the original WGAN model [5], and the second part includes the gradient penalty and its coefficient λ. Here, x and $\tilde{x}$ are the natural 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 [21] and this allows training for a higher number of iterations. Here, we train our model for 3,000 epochs.

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

Expression Prediction Optimization The Xpresso model can predict the expression of a one-hot encoded DNA sequence with a fixed length of 10,500 nucleotides. The model receives sequences of 3,500 nucleotides downstream and 7,000 nucleotides upstream of a Transcription Start Site (TSS) and predicts the log TPM expression for the sequences.

Let z_0 denote the initial latent vector, then the generator inputs the latent vector $(G(z_0))$ and generates the 5' UTR, which we call U_g. S denotes the gene sequence for which we design the 5' UTR with the original 5' UTR U_r. The expression prediction model is denoted as $E(S)$ where S is the input sequence.

To carry out the optimization, the initial 5' UTRs are first generated using a fixed seed for the random latent vector. They are attached to the selected gene sequences replacing the original 5' UTRs, and the initial expression is measured. We define R as the function to replace the original 5' UTR U_r with the generated 5' UTR U_G while maintaining the length of S and the TSS position on the sequence (see Equation 2). The original position of the TSS in the DNA samples is the 7000th nucleotide from the start. The length of a sequence is obtained via the *len* function. We use the bracket notation to slice the sequences.

$$\begin{aligned} R(U_g, S) = S[0:7000] + U_g + \\ S[7000 + len(U_g) : (10500 - (len(U_g) - len(U_r)))] \end{aligned} \tag{2}$$

To perform the Gradient Ascent to the input latent vector for optimization, we first calculate the gradient of the output of the Xpresso model with respect to its input DNA sequences (Equation 3) and slice the part of the gradient that corresponds to the 5' UTR from the 7000th to up to 7128th index where the 5' UTR is located depending on L which is the length of the generated 5' UTR. In the next step, the gradient of the generator is calculated with respect to the input noise vector. Updating the noise vector according to the gradient is supposed to increase the expression in the following iteration. This iteration is repeated 3,000 times by updating the input latent vector (see Equation 4). We store the history of predicted expression values for each iteration and pick the best-performing sequence as the optimized version. Although the optimization is performed as shown in Equation 4 on the batch of sequences, the final sequences are selected independently

as if the same optimization was performed separately for each element of the latent vector. This is to enable batch optimization instead of performing optimizations on individual UTRs separately.

$$\frac{\partial E(R(G(z_0),S))}{\partial z_0} = \frac{\partial E(R(G(z_0),S))}{\partial R(G(z_0),S)}[7000:7000+L] \times \frac{\partial G(z_0)}{\partial z_0} \tag{3}$$

$$z_{new} = z_0 + \frac{\partial E(R(G(z_0),S))}{\partial z_0} \tag{4}$$

Mean Ribosome Load and Translation Efficiency Prediction Optimization The FramePool model was trained on UTR sequences of lengths up to 100 bp. It utilizes the Frame-slice layer which allows it to predict the MRL value for 5' UTR sequences of any length [23]. Frame-slicing slices the sequence into 3-mers or codons referred to as frames. It applies global pooling on the indices, which lets it work with sequences of any size. To perform the optimization with this model, again, we use the Gradient Ascent algorithm. Here, slicing is not required as the model only inputs the UTR, and the derivatives are taken directly on the 5' UTR sequence. Considering the MRL prediction model as M and the initial latent vector as z_0 again, the gradient is calculated as shown in Equation 5 and applied as shown in Equation 6. This is repeated for 10,000 iterations and then the optimized sequences are selected as done for expression maximization and explained in Section 4.2. We also use MTtrans *3R* for translation efficiency optimization, a multi-task learning model trained on 3 ribosome profiling datasets. The optimization is performed similarly with a maximum of 10,000 iterations and the best-performing sequences are selected as mentioned above.

$$\frac{\partial M(G(z_0))}{\partial z_0} = \frac{\partial M(G(z_0))}{\partial G(z_0)} \times \frac{\partial G(z_0)}{\partial z_0} \tag{5}$$

$$z_{new} = z_0 + \frac{\partial M(G(z_0))}{\partial z_0} \tag{6}$$

Competing Interests: The authors declare no competing interests.

Author Contributions: SB and FO designed and implemented the computational model. SB, AH, 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 UTRdb 2.0 database [29]. The Human gene sequences are exported from Ensemble Biomart [14].

Code Availability: The source code, including the implementation of the model, the optimization procedure, and the additional scripts used to generate plots are released at http://github.com/ciceklab/UTRGAN.

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. Agarwal, V., Shendure, J.: Predicting mrna abundance directly from genomic sequence using deep convolutional neural networks. Cell reports **31**(7), 107663 (2020)
3. Andronescu, M., Fejes, A.P., Hutter, F., Hoos, H.H., Condon, A.: A new algorithm for rna secondary structure design. Journal of molecular biology **336**(3), 607–624 (2004)
4. Arhondakis, S., Clay, O., Bernardi, G.: Gc level and expression of human coding sequences. Biochemical and Biophysical Research Communications **367**(3), 542–545 (2008)
5. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International conference on machine learning. pp. 214–223. PMLR (2017)
6. Benson, L.N., Liu, Y., Deck, K., Mora, C., Mu, S.: Ifn-γ contributes to the immune mechanisms of hypertension. Kidney360 **3**(12), 2164–2173 (2022)

7. Burkhardt, D.H., Subramanian, R.R., Cobaugh, C.: Heterologous utr sequences for enhanced mrna expression (Jul 19 2022), uS Patent 11,389,546
8. Busch, A., Backofen, R.: Info-rna—a server for fast inverse rna folding satisfying sequence constraints. Nucleic acids research **35**(suppl_2), W310–W313 (2007)
9. 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)
10. Cao, J., Novoa, E.M., Zhang, Z., Chen, W.C., Liu, D., Choi, G.C., Wong, A.S., Wehrspaun, C., Kellis, M., Lu, T.K.: High-throughput 5 utr engineering for enhanced protein production in non-viral gene therapies. Nature communications **12**(1), 4138 (2021)
11. Castillo-Hair, S.M., Seelig, G.: Machine learning for designing next-generation mrna therapeutics. Accounts of Chemical Research **55**(1), 24–34 (2021)
12. Chen, C.Y., Tran, D.M., Cavedon, A., Cai, X., Rajendran, R., Lyle, M.J., Martini, P.G., Miao, C.H.: Treatment of hemophilia a using factor viii messenger rna lipid nanoparticles. Molecular Therapy-Nucleic Acids **20**, 534–544 (2020)
13. Cliff, N.: Dominance statistics: Ordinal analyses to answer ordinal questions. Psychological bulletin **114**(3), 494 (1993)
14. Cunningham, F., Allen, J.E., Allen, J., Alvarez-Jarreta, J., Amode, M., Armean, I., Austine-Orimoloye, O., Azov, A., Barnes, I., Bennett, R., Berry, A., Bhai, J., Bignell, A., Billis, K., Boddu, S., Brooks, L., Charkhchi, M., Cummins, C., Da Rin Fioretto, L., Davidson, C., Dodiya, K., Donaldson, S., El Houdaigui, B., El Naboulsi, T., Fatima, R., Giron, C.G., Genez, T., Martinez, J., Guijarro-Clarke, C., Gymer, A., Hardy, M., Hollis, Z., Hourlier, T., Hunt, T., Juettemann, T., Kaikala, V., Kay, M., Lavidas, I., Le, T., Lemos, D., Marugán, J.C., Mohanan, S., Mushtaq, A., Naven, M., Ogeh, D., Parker, A., Parton, A., Perry, M., Piližota, I., Prosovetskaia, I., Sakthivel, M., Salam, A., Schmitt, B., Schuilenburg, H., Sheppard, D., Pérez-Silva, J., Stark, W., Steed, E., Sutinen, K., Sukumaran, R., Sumathipala, D., Suner, M.M., Szpak, M., Thormann, A., Tricomi, F.F., Urbina-Gómez, D., Veidenberg, A., Walsh, T., Walts, B., Willhoft, N., Winterbottom, A., Wass, E., Chakiachvili, M., Flint, B., Frankish, A., Giorgetti, S., Haggerty, L., Hunt, S., IIsley, G., Loveland, J., Martin, F., Moore, B., Mudge, J., Muffato, M., Perry, E., Ruffier, M., Tate, J., Thybert, D., Trevanion, S., Dyer, S., Harrison, P., Howe, K., Yates, A., Zerbino, D., Flicek, P.: Ensembl 2022. Nucleic Acids Research **50**(D1), D988–D995 (11 2021). https://doi.org/10.1093/nar/gkab1049, `https://doi.org/10.1093/nar/gkab1049`
15. Feldman, R.A., Fuhr, R., Smolenov, I., Ribeiro, A.M., Panther, L., Watson, M., Senn, J.J., Smith, M., Almarsson, , Pujar, H.S., et al.: mrna vaccines against h10n8 and h7n9 influenza viruses of pandemic potential are immunogenic and well tolerated in healthy adults in phase 1 randomized clinical trials. Vaccine **37**(25), 3326–3334 (2019)
16. 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)
17. Frogner, C., Zhang, C., Mobahi, H., Araya, M., Poggio, T.A.: Learning with a wasserstein loss. Advances in neural information processing systems **28** (2015)
18. Gebre, M.S., Rauch, S., Roth, N., Yu, J., Chandrashekar, A., Mercado, N.B., He, X., Liu, J., McMahan, K., Martinot, A., et al.: Optimization of non-coding regions for a non-modified mrna covid-19 vaccine. Nature **601**(7893), 410–414 (2022)
19. Gillmore, J.D., Gane, E., Taubel, J., Kao, J., Fontana, M., Maitland, M.L., Seitzer, J., O'Connell, D., Walsh, K.R., Wood, K., et al.: Crispr-cas9 in vivo gene editing for transthyretin amyloidosis. New England Journal of Medicine **385**(6), 493–502 (2021)
20. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
21. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. Advances in neural information processing systems **30** (2017)
22. Hofacker, I.L., Fontana, W., Stadler, P.F., Bonhoeffer, L.S., Tacker, M., Schuster, P., et al.: Fast folding and comparison of rna secondary structures. Monatshefte fur chemie **125**, 167–167 (1994)
23. Karollus, A., Avsec, Ž., Gagneur, J.: Predicting mean ribosome load for 5'utr of any length using deep learning. PLoS computational biology **17**(5), e1008982 (2021)
24. Konu, Ö., Li, M.D.: Correlations between mrna expression levels and gc contents of coding and untranslated regions of genes in rodents. Journal of Molecular Evolution **54**(1), 35–41 (2002)
25. Kudla, G., Lipinski, L., Caffin, F., Helwak, A., Zylicz, M.: High guanine and cytosine content increases mrna levels in mammalian cells. PLoS biology **4**(6), e180 (2006)
26. Lee, Y.J., Seo, H.W., Baek, J.H., Lim, S.H., Hwang, S.G., Kim, E.H.: Gene expression profiling of glioblastoma cell lines depending on tp53 status after tumor-treating fields (ttfields) treatment. Scientific reports **10**(1), 1–14 (2020)
27. 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)
28. Liang, Y., Huang, L., Liu, T.: Development and delivery systems of mrna vaccines. Frontiers in Bioengineering and Biotechnology **9** (2021)

29. Lo Giudice, C., Zambelli, F., Chiara, M., Pavesi, G., Tangaro, M.A., Picardi, E., Pesole, G.: Utrdb 2.0: a comprehensive, expert curated catalog of eukaryotic mrnas untranslated regions. Nucleic Acids Research **51**(D1), D337–D344 (2023)
30. 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)
31. Lu, T.K.T., Kellis, M., Cao, J., Pardo, E.M.N., Zhang, Z.: Synthetic 5'utr sequences, and high-throughput engineering and screening thereof (Feb 27 2020), uS Patent App. 16/441,647
32. 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)
33. Noreen, M., Arshad, M.: Association of tlr1, tlr2, tlr4, tlr6, and tirap polymorphisms with disease susceptibility. Immunologic Research **62**(2), 234–252 (2015)
34. Ozaki, T., Nakagawara, A.: Role of p53 in cell death and human cancers. Cancers **3**(1), 994–1013 (2011)
35. Ozden, F., Barazandeh, S., Akboga, D., Seker, U.O.S., Cicek, A.E.: Rnagen: A generative adversarial network-based model to generate synthetic rna sequences to target proteins. bioRxiv pp. 2023–07 (2023)
36. Pardi, N., Hogan, M.J., Pelc, R.S., Muramatsu, H., Andersen, H., DeMaso, C.R., Dowd, K.A., Sutherland, L.L., Scearce, R.M., Parks, R., et al.: Zika virus protection by a single low-dose nucleoside-modified mrna vaccination. Nature **543**(7644), 248–251 (2017)
37. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
38. Robinson, E., MacDonald, K.D., Slaughter, K., McKinney, M., Patel, S., Sun, C., Sahay, G.: Lipid nanoparticle-delivered chemically modified mrna restores chloride secretion in cystic fibrosis. Molecular Therapy **26**(8), 2034–2046 (2018)
39. Rohner, E., Yang, R., Foo, K.S., Goedel, A., Chien, K.R.: Unlocking the promise of mrna therapeutics. Nature biotechnology **40**(11), 1586–1600 (2022)
40. Sahin, U., Oehm, P., Derhovanessian, E., Jabulowsky, R.A., Vormehr, M., Gold, M., Maurus, D., Schwarck-Kokarakis, D., Kuhn, A.N., Omokoko, T., et al.: An rna vaccine drives immunity in checkpoint-inhibitor-treated melanoma. Nature **585**(7823), 107–112 (2020)
41. Sample, P.J., Wang, B., Reid, D.W., Presnyak, V., McFadyen, I.J., Morris, D.R., Seelig, G.: Human 5 utr design and variant effect prediction from a massively parallel translation assay. Nature biotechnology **37**(7), 803–809 (2019)
42. Sav, S., Hampson, D.J., Tsang, H.H.: Simard: A simulated annealing based rna design algorithm with quality pre-selection strategies. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1–8. IEEE (2016)
43. Sultana, N., Hadas, Y., Sharkar, M.T.K., Kaur, K., Magadum, A., Kurian, A.A., Hossain, N., Alburquerque, B., Ahmed, S., Chepurko, E., et al.: Optimization of 5 untranslated region of modified mrna for use in cardiac or hepatic ischemic injury. Molecular Therapy-Methods & Clinical Development **17**, 622–633 (2020)
44. Sun, Y., Hu, B., Stanley, G., Harris, Z.M., Gautam, S., Homer, R., Koff, J.L., Rajagopalan, G.: Interferon-γ is protective in cytokine release syndrome-associated extrapulmonary acute lung injury. American Journal of Respiratory Cell and Molecular Biology (2022)
45. Trotta, E.: On the normalization of the minimum free energy of rnas by sequence length. PloS one **9**(11), e113380 (2014)
46. Ulmer, J.B., Geall, A.J.: Recent innovations in mrna vaccines. Current opinion in immunology **41**, 18–22 (2016)
47. Vinh, L.V., Lang, T.V., Binh, L.T., Hoai, T.V.: A two-phase binning algorithm using l-mer frequency on groups of non-overlapping reads. Algorithms for Molecular Biology **10**(1), 1–12 (2015)
48. Vogel, A.B., Kanevsky, I., Che, Y., Swanson, K.A., Muik, A., Vormehr, M., Kranz, L.M., Walzer, K.C., Hein, S., Güler, A., et al.: Bnt162b vaccines protect rhesus macaques from sars-cov-2. Nature **592**(7853), 283–289 (2021)
49. Von Niessen, A.O., Fesser, S., Vallazza, B., Beissert, T., Kuhn, A., Sahin, U., Poleganov, M.A.: 3'-utr sequences for stabilization of rna (Mar 7 2019), uS Patent App. 15/763,709
50. Von Niessen, A.O., Fesser, S., Vallazza, B., Beissert, T., Kuhn, A., Sahin, U., Poleganov, M.A.: 3-utr sequences for stabilization of rna (Nov 8 2022), uS Patent 11,492,628
51. Zheng, W., Fong, J.H., Wan, Y.K., Chu, A.H., Huang, Y., Wong, A.S., Ho, J.W.: Translation rate prediction and regulatory motif discovery with multi-task learning. In: International Conference on Research in Computational Molecular Biology. pp. 139–154. Springer (2023)
52. Zrimec, J., Buric, F., Kokina, M., Garcia, V., Zelezniak, A.: Learning the regulatory code of gene expression. Frontiers in Molecular Biosciences **8** (2021)
53. Zrimec, J., Buric, F., Muhammad, A.S., Chen, R., Verendel, V., Töpel, M., Zelezniak, A.: Gene expression is encoded in all parts of a co-evolving interacting gene regulatory structure. bioRxiv p. 792531 (2019)
54. Zrimec, J., Fu, X., Muhammad, A.S., Skrekas, C., Jauniskis, V., Speicher, N.K., Börlin, C.S., Verendel, V., Chehreghani, M.H., Dubhashi, D., et al.: Supervised generative design of regulatory dna for gene expression control. BioRxiv (2021)