Designing Biological Sequences without Prior Knowledge Using Evolutionary Reinforcement Learning

Xi Zeng[1], Xiaotian Hao[2], Hongyao Tang[2], Zhentao Tang[3], Shaoqing Jiao[1], Dazhi Lu[1], Jiajie Peng[1, 4, 5*]

[1]School of Computer Science, Northwestern Polytechnical University
[2]College of Intelligence and Computing, Tianjin University
[3]Noah's Ark Lab, Huawei
[4]Key Laboratory of Big Data Storage and Management, Northwestern Polytechnical University, Ministry of Industry and Information Technology
[5]School of Computer Science, Research and Development Institute of Northwestern Polytechnical University in Shenzhen
{xizeng,jiaosq22,dzlu}@mail.nwpu.edu.cn, {xiaotianhao,bluecontra}@tju.edu.cn, tangzhentao1@huawei.com, jiajiepeng@nwpu.edu.cn

Abstract

Designing novel biological sequences with desired properties is a significant challenge in biological science because of the extra large search space. The traditional design process usually involves multiple rounds of costly wet lab evaluations. To reduce the need for expensive wet lab experiments, machine learning methods are used to aid in designing biological sequences. However, the limited availability of biological sequences with known properties hinders the training of machine learning models, significantly restricting their applicability and performance. To fill this gap, we present ERLBioSeq, an Evolutionary Reinforcement Learning algorithm for BIOlogical SEQuence design. ERLBioSeq leverages the capability of reinforcement learning to learn without prior knowledge and the potential of evolutionary algorithms to enhance the exploration of reinforcement learning in the large search space of biological sequences. Additionally, to enhance the efficiency of biological sequence design, we developed a predictor for sequence screening in the biological sequence design process, which incorporates both the local and global sequence information. We evaluated the proposed method on three main types of biological sequence design tasks, including the design of DNA, RNA, and protein. The results demonstrate that the proposed method achieves significant improvement compared to the existing state-of-the-art methods.

Introduction

Designing biological sequences with desired properties is of great significance for both biology and chemistry (Bennett et al. 2023; Silva et al. 2019; Song and Li 2023; Wang et al. 2020). However, designing novel biological sequences with desired properties, such as binding affinity (Tinberg et al. 2013), antimicrobial activity (Torres et al. 2021), or stability (Mourtada et al. 2019), poses challenges due to the requirement of exploring a discrete and extensive search space (Jain et al. 2022). Artificial intelligence (AI) can assist humans in efficiently designing biological sequences, thus accelerating scientific discovery (Zrimec et al. 2022; Repecka et al. 2021). However, the deployment of AI may fail due to a lack of prior knowledge (Dama et al. 2023). Unfortunately, designing biological sequences without prior knowledge is valuable in diverse domains, such as designing regulatory DNA to control gene expression (Zrimec et al. 2022) and designing proteins to catalyze desired chemical reactions (Repecka et al. 2021). Therefore, it is important to develop a method for novel biological sequence design without prior knowledge.

Reinforcement learning (RL) can start without prior knowledge (Schrittwieser et al. 2020; Silver et al. 2018). Through training the agent with rewards, it becomes possible to devise a strategy capable of surpassing even a world-champion human player (Silver et al. 2021). In the field of biological science, under the constraint of limited data availability, Dama et al. (2023) employs RL in the mapping of microbial metabolism without prior knowledge. Moreover, RL has also been applied in the domain of biological sequence design (Angermueller et al. 2019). However, RL agents must balance known rewards with unfamiliar data, resulting in limited exploration of the biological sequence space and difficulty in finding the ideal sequence. As a result, applying RL to designing biological sequences with desired properties within a vast sequence space is not straightforward.

Evolutionary algorithms (EAs) are the more common approach for biological sequence design, recognized as the gold standard in this field (Arnold 1998). They have risen as formidable competitors for biological sequence design owing to their broad applicability (Sinai et al. 2020; Ren et al. 2022). However, the EAs focus on exploring the local space (Auger and Hansen 2005). The inefficient utilization of evolutionary directional information results in lower efficient searches for biological sequences that satisfy desired properties.

In biological sequence design, multiple rounds of experimentation are often needed (Jain et al. 2022). Directly applying generated sequences to wet experiments can be time-consuming and labor-intensive. To address the problem, cre-

*Corresponding author.
Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

ating a fitness prediction model for screening can significantly reduce the burden of wet experiments (Ren et al. 2022). Sinai et al. (2020) included an ensemble convolutional neural network in their robust evolutionary greedy algorithm. Additionally, Ren et al. (2022) proposed a Mutation Factorization Network architecture for sequence selection. However, these approaches either focus on particular situations or demand integration with existing models, ignoring the optimization of feature extraction from biological sequences for predictive models. The process of feature extraction is critical to the success of machine learning (Storcheus, Rostamizadeh, and Kumar 2015). Furthermore, the incorporation of more comprehensive features bears the capacity to enhance the precision of result predictions (Li et al. 2023).

In this work, we propose a novel framework called ERLBioSeq for the effective design of biological sequences without prior knowledge. We integrate the strengths of RL and evolutionary algorithm (EA) to design biological sequences. Specifically, to address the challenge of RL's limited exploration in complex sequence environments, we suggest employing EA to explore biological sequences produced by RL. This will enhance RL's exploratory capacity. Meanwhile, RL has the capability to incorporate populations of biological sequences as direction information into EA. This addresses the issue of EA's inefficient utilization of evolutionary directional information. We leverage these characteristics to facilitate mutual enhancement between the two algorithms. Additionally, we design a predictor that utilizes local and global information extracted from biological sequences as features for predicting sequence fitness. The predictor facilitates sequence screening in the biological sequence design process through the extraction of more comprehensive features. To the best of our knowledge, this is the first evolutionary reinforcement learning-based method for designing novel biological sequences. The major contributions of our work are outlined as follows:

- We introduce a methodology that combines the advantages of EA and RL, for the design of biological sequences.
- We propose a predictor that utilizes both local and global information derived from biological sequences as features to predict sequence fitness.
- The evaluation results demonstrate that ERLBioSeq outperforms state-of-the-art methods in discovering high-scoring candidates with fewer rounds of experimentation. By harnessing local and global information extracted from biological sequences, the predictor can enhance the performance of designing biological sequences.

Related Work

Evolutionary Algorithms. EAs are employed in designing biological sequences through iterative generation and evolution. This process is guided by fitness criteria to optimize the desired function. In detail, Directed Evolution, acknowledged with a Nobel Prize in 2018, currently stands as the esteemed gold standard for biomolecular design and employs a randomized local search approach (Arnold 1998). Sinai et al. (2020) presents a new algorithm called AdaLead, which is a simple and robust way of finding biological sequences for a given function. The algorithm employs a model to predict the function of a sequence, acting as an oracle. It then selects the suitable sequence based on the feedback from this oracle. Ren et al. (2022) present a new method for designing protein sequences. The method uses a model-guided exploration strategy that starts from a known sequence and iteratively modifies it by sampling from a distribution of proximal sequences. However, evolutionary algorithm-based methods exhibit limitations in terms of search efficiency within the expansive exponential search space, rendering them inadequate to fulfill the efficiency requirements associated with biological sequence design.

Generative Models. Generative models possess the capacity to simulate data distributions and create entirely novel biological sequences through sampling from these distributions. Gupta and Zou (2019) introduced feedback GAN (FBGAN), a new type of generative model that uses a feedback loop to adjust the generated sequences based on an external function analyzer. Brookes, Park, and Listgarten (2019) introduced a model-based adaptive sampling method that estimates a distribution over the design space based on desired properties. This method prevents the exploration algorithm from becoming trapped in regions of poor model generalization. Jain et al. (2022) applied an active learning algorithm that uses GFlowNets, which is a generative model to produce diverse and novel candidates. The method also incorporates existing labeled data to speed up learning. However, generative models exhibit a significant reliance on data, resulting in limited availability for designing novel biological sequences.

Reinforcement Learning. RL can learn biological sequence design without prior knowledge of biological sequences. Angermueller et al. (2019) proposed Dynappo, a model-based RL method that uses a proximal-policy optimization (PPO) (Schulman et al. 2017) algorithm to train a generative sequence model that can produce diverse and high-quality sequences. To improve sample efficiency, Dynappo also uses a surrogate model that approximates the experimental function based on previous observations. However, general RL lacks the exploration of the environment, resulting in low search efficiency.

Problem Formulation and Method

In this section, we first formulate the biological sequence design problem. After that, we will introduce the key components of the ERLBioSeq framework in detail.

Problem Formulation

The primary challenge in biological sequence design is to create a sequence $x \in V^T$ with the desired properties, where V is the set of amino acids ($|V| = 20$), DNA nucleotides ($|V| = 4$), or RNA nucleotides ($|V| = 4$), and T is the length of the sequence. The ultimate goal is to maximize the fitness of the sequence through the fitness landscape $f(x)$, where the fitness landscape characterizes the mapping between biological sequences and their functional levels. The

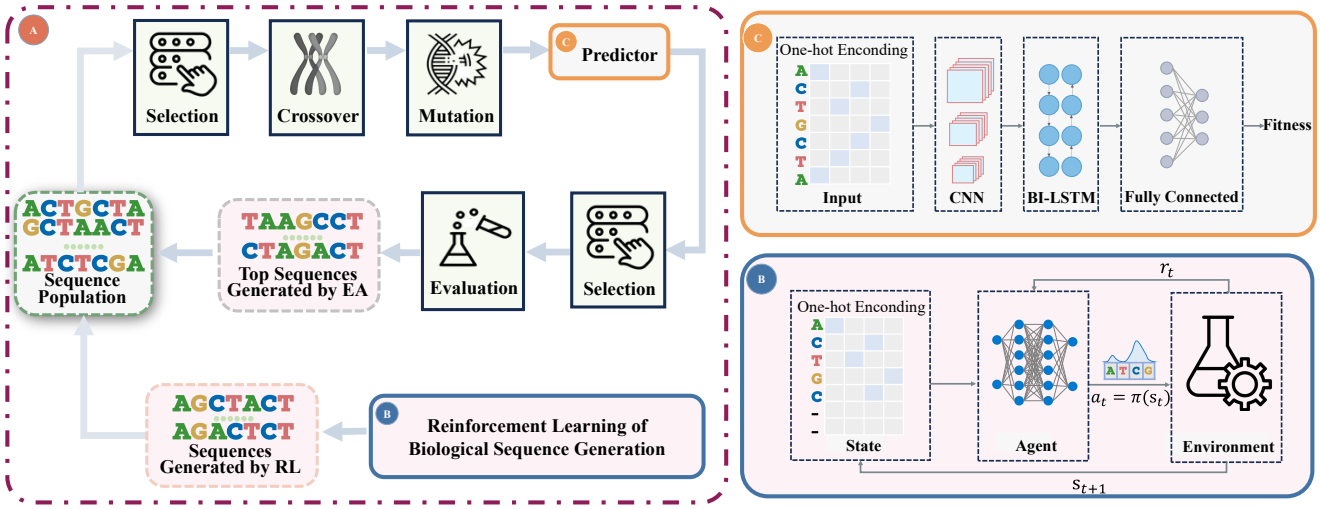

Figure 1: The main workflow of ERLBioSeq. (A) This figure shows how ERLBioSeq combines EA and RL in biological sequence design. (B) This diagram illustrates the process of employing RL for the design of biological sequences. (C) The architecture of fitness predictor.

design process encompasses N iterations. In each iteration i (where $i \in \{1, 2, ..., N\}$), a collection of sequences labeled as $\mathcal{X}_i$ is generated, where $\mathcal{X}_i = \{x_i^1, x_i^2, ..., x_i^B\}$. The number of queries of the fitness landscape $f(x)$ in each round is denoted as B.

Subsequently, the fitness of these sequences is determined by applying the fitness landscape $f(x)$ to each of the sequences, resulting in a collection of fitness values denoted as $\mathcal{Y}_i = \{y_i^1, y_i^2, ..., y_i^B\}$. These fitness values play a crucial role in constructing distinct datasets referred to as $\mathcal{D}_i$, which are individually generated in each design round. Formally, $\mathcal{D}_i$ is represented as $\{(x_i^1, y_i^1), (x_i^2, y_i^2), ..., (x_i^B, y_i^B)\}$. As the design process advances, the cumulative dataset $\mathcal{X}_{all}$, $\mathcal{Y}_{all}$, $\mathcal{D}_{all}$ is formed by merging the various sets $\mathcal{X}_i$, $\mathcal{Y}_i$, $\mathcal{D}_i$, respectively. One of our goals is to maximize the best score in $\mathcal{X}_{all}$, which can be formalized as:

$$Top(\mathcal{X}_{all}) = \arg\max_{x \in \mathcal{X}_{all}} f(x). \quad (1)$$

Beyond achieving the highest score, we also contemplate the mean value of the top-k biological sequences, because the sequence with the highest score might not align with practical requirements, necessitating additional candidate sequences for comprehensive testing.

Overview of ERLBioSeq

We first introduce the main workflow of our framework ERLBioSeq, as shown in Figure 1, we utilize the design of DNA sequences to describe our model's process. Figure 1A shows the algorithm how to combine EA and RL in biological sequence design, in each round of design, sequences obtained from both the EA and RL are merged. New sequences are generated by the EA through selection, crossover, and mutation mechanisms. Following that, the novel sequences are evaluated and filtered according to fitness predictor scores. Concurrently, RL will generate biological sequences with desired properties. These sequences were subsequently integrated into sequence populations for the next optimization round. Figure 1B illustrates the process of employing RL for the design of biological sequences. For DNA sequences, the agent performs an action a_t given the present state s_t, gains a reward r_t, and iterates the subsequent state update until the designated sequence length is attained. Figure 1C is a biological sequence fitness predictor that utilizes both local and global features, to augment the filtration of sequences proposed by the EA. Specifically, we extract local features from biological sequences using CNN, and global features using BI-LSTM, to facilitate the prediction of biological sequence fitness. Next, we will introduce each module in detail.

Combining Evolutionary and Reinforcement Learning for Biological Sequence Design

In this section, we introduce how to combine EA and RL in biological sequence design. RL can learn online without prior knowledge but lacks exploration of the environment. Classical EA relies on random mutation and a combination of populations, which have wide applicability and stability but they suffer from low sample efficiency and a weaker ability to harness evolutionary directional information. We consider combining EA and RL algorithms for the design of biological sequences, where RL can supply directional information through population sequences for EA, and EA can enhance the exploration ability of RL and help RL search and find better sequences in local space, RL and EA mutually enhance performance.

Reinforcement Learning to Design Biological Sequences. The process of designing a sequence using RL can be formulated as a Markov decision process (MDP) like $M =$

$\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ represents the set of states, while $\mathcal{A}$ denotes the set of actions, $\mathcal{P}$ refers to the transition function, $\mathcal{R}$ represents the set of rewards associated with being in a particular state and taking a specific action, $\gamma \in [0, 1)$ is the discount factor. A policy $\pi \in P(\mathcal{A})^{|\mathcal{S}|}$ defines the distribution over all action for each state. The goal of RL is to find an optimal policy π^* that maximizes the expected long-term discounted return.

For biological sequence design issues, a sequence is generated sequentially from left to right. At time step t, state equation $s_t = \{a_0, a_1, ..., a_{t-1}\}$, and action $a_t \in \mathcal{A}$ is executed to reach the next state, where a_t is an amino acid or a nucleotides. The transition function $P(s_{t+1}|s_t)$ is deterministic, so a_t is directly added to the original sequence. Except for the reward in the last step, the reward in the other steps is 0, and the reward is related to $f(s_T)$. The discounted return can be formalized as:

$$G(\tau) = \sum_{t=0}^{T-1} \gamma^t r_{t+1}. \tag{2}$$

Where τ represents a trajectory where a complete sequence is generated, we trained a policy π_θ to optimize the total expected rewards:

$$\mathcal{J}(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)}[G(\tau)] = \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_{t=0}^{T-1} \gamma^t r_{t+1} \right]. \tag{3}$$

The policy takes as input the current state s_t and outputs an action a_t that maximizes the expected reward overall future time steps. The workflow of RL to design biological sequences is shown in Figure 1B. We use a one-hot vector encoding state, if there is no amino acid or nucleotides at this position, then the encoding vector of this position is all 0.

Evolutionary Algorithm for Designing Biological Sequences. The evolutionary algorithm starts with a random initial set of sequences, denoted as $\mathcal{X}_{random}$. During the i-th ($i > 1$) design round, a subset of high-performing sequences was chosen from D_{all} based on their fitness scores. This process can be defined as follows:

$$R = \left\{ x | f(x) \geq \max_{y \in \mathcal{Y}_{all}} y \cdot (1 - \kappa), \forall x \in \mathcal{X}_{all} \right\}. \tag{4}$$

Here, κ serves as the threshold governing the fraction of screened sequences. These sequences undergo recombination and mutation processes to produce novel mutants. These mutants are then assessed using a fitness predictor denoted as $f'(x)$, which quantifies their quality. Following experimental validation of sequences filtered by their fitness scores, a novel dataset D_i is obtained. These sequences are preserved within the candidate library $\mathcal{D}_{all}$ for subsequent exploration. This procedure generally involves several iterations of optimization, each designed to iteratively enhance the population sequence.

Combining Evolutionary Algorithms and Reinforcement Learning. To utilize the benefits of both methods, we propose the ERLBioSeq framework, which combines RL and EA to generate new sequences that meet the desired properties. The biological sequences obtained from both algorithms are integrated into the sequence population for further optimization. Algorithm 1 summarizes the overall procedure of our algorithm. To generate the candidate batch, we use a design algorithm that combines EA and RL. Specifically, with the exception of the initial design round that employed random sequences $\mathcal{X}_{random}$, in subsequent rounds we extract a subset of sequences from $\mathcal{D}_{all}$, and then apply $MutationAndCrossover$ (detail in Appendix A.1[1]) operations to these sequences to produce $v \cdot B$ new sequences M, where v is used to adjust the number of biological sequences generated by each round of EA.

Then, we use the fitness model f' to select the best $B/2$ sequences to S from M and measured their fitness using fitness landscape $f(x)$ to get $\mathcal{D}_{evo}$. We also use the RL policy π_θ to generate $B/2$ sequences for $\mathcal{D}_{rl}$. We add the sequences from $\mathcal{D}_{evo}$ and $\mathcal{D}_{rl}$ to $\mathcal{D}_{all}$, and use $\mathcal{D}_{all}$ to train the fitness model f' and $\mathcal{D}_{rl}$ to train the RL policy π_θ. It consists of N rounds of interactions for continued iteration.

Fitness Model Design. In the framework of ERLBioSeq, we have a fitness model f' for screening sequences, this is a challenge in designing biological sequences (Ren et al. 2022). Existing studies such as ensemble CNN (Sinai et al. 2020), Mutation Factorization Network (Ren et al. 2022), etc., either focus on specific scenarios or need to be inte-

Algorithm 1: ERLBioSeq

Input: experiment rounds N, fitness model f', batch B, threshold κ, scaling factor v, RL policy π_θ, buffer $\mathcal{D}_{rl}$, $\mathcal{D}_{evo}$, $\mathcal{D}_{all}$, mutants M, random sequences $\mathcal{X}_{random}$
Initialize : $\mathcal{D}_{all} \leftarrow \emptyset$

for $i = 1, 2, 3...N$ **do**
  Let $\mathcal{D}_{rl} \leftarrow \emptyset$, $\mathcal{D}_{evo} \leftarrow \emptyset$, $M \leftarrow \emptyset$
  if $i > 1$ **then**
    Form sets $\mathcal{X}_{all}$ and $\mathcal{Y}_{all}$ from the x-values and y-values in $\mathcal{D}_{all}$, respectively
    $R = \{x | f(x) \geq \max_{y \in \mathcal{Y}_{all}} y \cdot (1 - \kappa), \forall x \in \mathcal{X}_{all}\}$
  else
    $R = \mathcal{X}_{random}$
  end if
  while $|M| < v \cdot B$ **do**
    $M = M \cup MutationAndCrossover(R)$
  end while
  Use f' to select S, the top $B/2$ sequence from M
  for $x \in S$ **do**
    $D_{evo} = D_{evo} \cup \{x, f(x)\}$
  end for
  while $|\mathcal{D}_{rl}| < B/2$ **do**
    Collect samples $\mathcal{D}_{rl} = \{x, f(x)\}$ using policy π_θ
  end while
  $\mathcal{D}_{all} = \mathcal{D}_{all} \cup \mathcal{D}_{rl} \cup \mathcal{D}_{evo}$
  Train f' with data from $\mathcal{D}_{all}$
  Train policy π_θ on $\mathcal{D}_{rl}$
end for

[1]https://github.com/reset001/ERLBioSeqappendix.

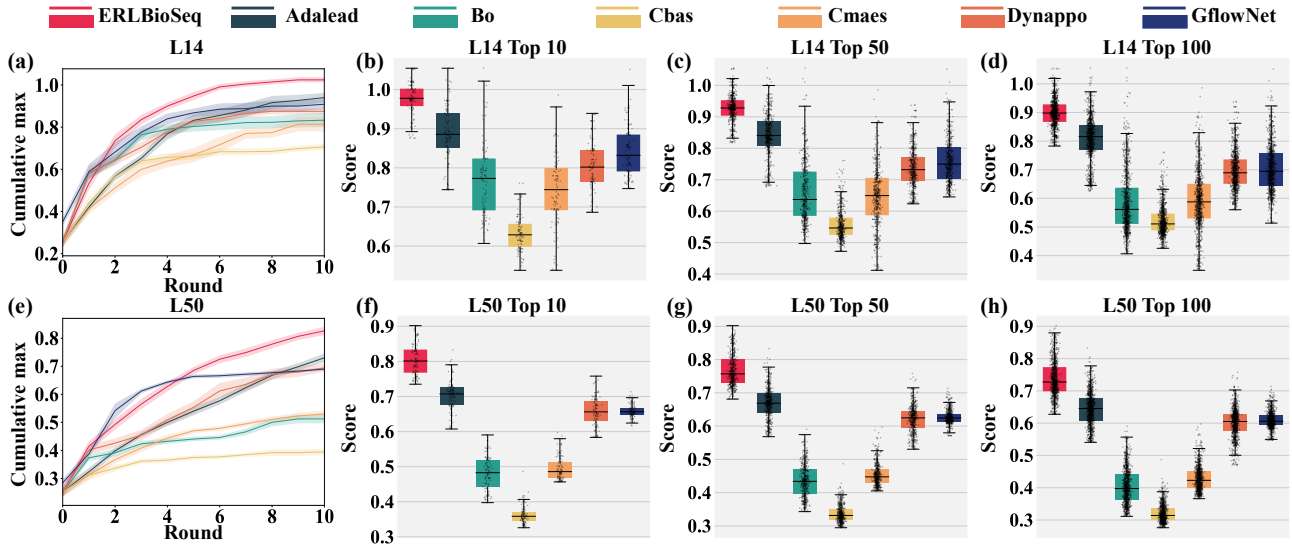

Figure 2: Comparison of methods on RNA design task. (a) and (e) are the relationship between the cumulative maximum value of the objective function $f(x)$ and the iteration round for RNAs of length 14 and 50, respectively, the shaded area is the standard deviation. (b-d) and (f-h) show the boxplots with data scatter distribution of RNA sequences with lengths of 14 and 50 in the top 10, 50, and 100 biological sequences, respectively. More experimental results on RNA design are in Appendix B.1.

grated with existing models. They ignore optimizing feature extraction for predictive models, a crucial process for successful machine learning (Storcheus, Rostamizadeh, and Kumar 2015). Moreover, integrating more comprehensive features can boost the accuracy of result predictions (Li et al. 2023). To address the challenge of inadequate feature extraction, we have developed a model, named Local Global Feature Extraction Network (LGFEN), which utilizes local and global biological sequence information in features to glean more comprehensive insights to predict the fitness score of biological sequences.

Figure 1C illustrates the structure of LGFEN. When provided with a biological sequence, our first step involves representing it through one-hot encoding. Inspired by the multiple windows scanning techniques in convolutional neural networks proposed by Ho, Le, and Ou (2022), we apply convolutional filters with varying window sizes to the input feature matrix, acquiring feature representations for diverse neighborhoods within the biological sequence. Currently, CNN has been employed to encode the feature vector of the biological sequence as $V = \{v_1, v_2, \dots v_T\}$, enhancing the localized information at each position t within the sequence. Nevertheless, v_t does not encompass the global information of the sequence. Long short-term memory networks (LSTM) (Hochreiter and Schmidhuber 1997) have been devised to capture long-range dependencies and encode sequential information. At time step t, the forward LSTM produces a vector $\overrightarrow{h_t}$, while the backward LSTM generates a vector $\overleftarrow{h_t}$. Through concatenating the outcomes of the left and right LSTM, we derive the BI-LSTM output $H_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$. Subsequently, a fully connected neural network is utilized to map the feature vector onto the fitness score y. The model is trained to employ the mean squared error loss as the objective function.

Experiments

In this section, we show experimental results from a range of biologically relevant sequence design tasks to demonstrate the effectiveness of our proposed ERLBioSeq algorithm. Additionally, we conduct ablation studies to explore the contribution of each individual design component.

Tasks and Evaluation Criteria

Following prior works, we choose some energy models and datasets to simulate the fitness landscape.

RNA Binding Task. This task aims to optimize RNA sequences to achieve the highest binding energy with nucleotide targets of lengths 14 and 50. The ViennaRNA package is utilized to compute the binding energy of RNA sequences (Lorenz et al. 2011). We follow the design task presented by Sinai et al. (2020). The size of search space is 4^{14} and 4^{50}.

Protein Design Task. We evaluate the algorithms in the context of protein design tasks, employing PyRosetta (Chaudhury, Lyskov, and Gray 2010) as the objective function. The objective function provides a scaled estimation of folding energy, reflecting the likelihood of sequence folding into the intended structure (Kuhlman et al. 2003). Adhering to the experimental configuration outlined in (Sinai et al. 2020), we optimize the structure of 3MSI, a 66-amino-acid antifreeze protein naturally occurring in oceanic environments (DeLuca et al. 1998). The size of the search space is 20^{66}.

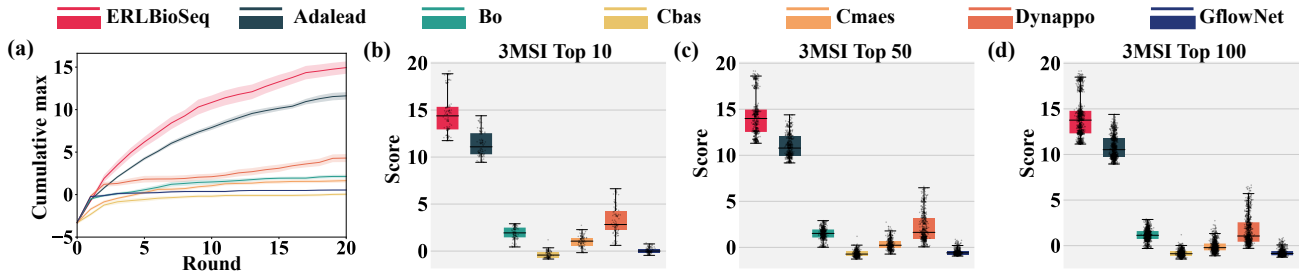

Figure 3: Comparison of methods on a protein design task. (a) is the relationship between the cumulative maximum value of the objective function $f(x)$ and the iteration round, the shaded area is the standard deviation. (b-d) show the boxplots with data scatter distribution in the top 10, 50, and 100. More experimental results on protein design are in Appendix B.2.

	ERLBioSeq	Adalead	Bo	Cbas	Cmaes	Dynappo	GflowNet
Top10 mean	**0.989±0.006**	0.984±0.007	0.958±0.021	0.974±0.013	0.967±0.022	0.937±0.026	0.870±0.071
Top50 mean	**0.970±0.013**	0.963±0.016	0.878±0.060	0.928±0.034	0.882±0.066	0.863±0.052	0.735±0.113
Top100 mean	**0.953±0.022**	0.941±0.026	0.801±0.094	0.886±0.053	0.786±0.116	0.800±0.079	0.629±0.159

Table 1: Mean binding activity of top 10, 50, and 100 DNA and transcription factor targets.

TF Bind 8 Task. This task aims to find DNA sequences of length 8 that have high binding activity to human transcription factors. We use the same data as in Barrera et al. (2016) and follow the experimental setup of Trabucco et al. (2022). The search space size is 4^8. Although the problem features a small search space, it is ideal for computer benchmarks due to its exhaustiveness, eliminating the need for estimating the missing $f(x)$.

Baselines

- **GflowNet** (Jain et al. 2022) is a generative model for diverse and novel biological sequences (e.g., proteins, DNA) with desired properties. It employs flows to capture complex dependencies and constraints of these sequences.
- **AdaLead** (Sinai et al. 2020) implements model-guided evolution, performing hill-climbing on high-fitness query sequences in each batch round.
- **Dynappo** (Angermueller et al. 2019) treats biological sequence design as a sequential decision problem, using model-based RL with proximal policy optimization (Schulman et al. 2017) to learn search strategies.
- **Cbas** (Brookes, Park, and Listgarten 2019) confines the sampling distribution, resulting in a trust region search with the learned model.
- **Cmaes** (Hansen 2006), an established evolutionary search algorithm, adapts search strategies using covariance matrix estimation for the next generation.
- **Bayesian Optimization (Bo)** (Močkus 1975) is a classical approach to sequence design problems. We used the implementation developed by (Sinai et al. 2020).

Performance Comparison

We evaluated the performance of ERLBioSeq on problems involving DNA, RNA, and proteins. To enhance the robustness of our evaluation, each of our experimental outcomes represents 10 independent runs. Further experimental details are provided in Appendix A.2. Figures 2 (a), 2 (e) and 3 (a) depict the relationship between the cumulative maximum value of the objective function $f(x)$ and the iteration round for RNAs and protein 3MSI. Figures 2 (b-d), 2 (f-h) and 3 (b-d) are the box plots with data scatter distribution depicting the distribution of the top 10, 50, and 100 sequences of RNA and protein generated by various algorithms. Because the DNA binding design problem is relatively easy to optimize, the gap between various algorithms is not very large, so we use the top-k average method to compare various algorithms. Table 1 provides a summary of the outcomes derived from this comparison. Based on all the results, it is evident that ERLBioSeq outperforms the baseline algorithm. As Adalead is classified as EA and Dynappo is classified as RL, our implementation encompasses both algorithms. Consequently, our algorithm can be readily compared with these methodologies to conduct ablation experiments focusing on the components of EA and RL. Compared to the independent EA method Adalead and the RL method Dynappo, our results demonstrate that the combination of EA and RL outperforms any single algorithm. Implementation specifics of ERLBioSeq are provided in Appendix A.1.

Effectiveness of LGFEN

The effectiveness of biological sequence design is also influenced by the fitness prediction model's quality. To conduct an ablation study of the model, we explored four additional model architectures: CNN, MLP, LSTM, and BI-LSTM. The diverse model structures were applied to DNA, RNA, and protein design tasks. The corresponding experimental outcomes are presented in Table 2. The assessment scores represent the average performance scores of the top 10, 50, and 100 sequences produced by varying model structures. The outcomes demonstrate that LGFEN outperforms

		LGFEN	MLP	CNN	LSTM	BI-LSTM
Top10 mean	DNA	**0.989 ± 0.006**	0.981 ± 0.008	0.987 ± 0.007	0.970 ± 0.009	0.973 ± 0.007
	RNA14	**0.976 ± 0.043**	0.969 ± 0.038	0.942 ± 0.054	0.931 ± 0.057	0.964 ± 0.056
	RNA50	**0.805 ± 0.043**	0.728 ± 0.028	0.713 ± 0.040	0.736 ± 0.045	0.749 ± 0.042
	Protein	**14.621±2.218**	13.124±1.331	14.064±2.429	12.477±1.829	13.124±1.663
Top50 mean	DNA	**0.970 ± 0.013**	0.957 ± 0.019	0.965 ± 0.017	0.926± 0.033	0.939 ± 0.027
	RNA14	**0.927 ± 0.045**	0.915 ± 0.047	0.893 ± 0.057	0.872± 0.054	0.900 ± 0.059
	RNA50	**0.768 ± 0.048**	0.688 ± 0.035	0.677 ± 0.043	0.689± 0.052	0.704 ± 0.046
	Protein	**14.239±2.206**	12.767±1.285	13.679±2.418	12.089±1.789	12.691±1.658
Top100 mean	DNA	**0.953 ± 0.022**	0.932 ± 0.030	0.943 ± 0.028	0.889± 0.047	0.904 ± 0.043
	RNA14	**0.899 ± 0.049**	0.886 ± 0.051	0.868 ± 0.058	0.837± 0.058	0.866 ± 0.063
	RNA50	**0.737 ± 0.055**	0.660 ± 0.041	0.650 ± 0.048	0.656± 0.058	0.670 ± 0.053
	Protein	**14.020±2.204**	12.583±1.282	13.449±2.410	11.868±1.770	12.466±1.647

Table 2: Comparison of mean values of top 10, 50, and 100 biological sequences with different predictors in DNA, RNA, and protein design tasks.

		ERLBioSeq	EVO$_{Bo}$	EVO$_{Cmaes}$	EVO$_{Cbas}$	EVO$_{GflowNet}$
Top10 mean	DNA	**0.989 ± 0.006**	0.986 ± 0.006	0.987 ± 0.008	0.987 ± 0.007	0.988 ± 0.007
	RNA14	**0.976 ± 0.043**	0.943 ± 0.054	0.930 ± 0.058	0.902 ± 0.066	0.968 ± 0.041
	RNA50	**0.805 ± 0.043**	0.796 ± 0.04	0.762 ± 0.048	0.804 ± 0.038	0.784 ± 0.050
	Protein	**14.621±2.218**	12.294±1.007	11.821±1.296	11.907±1.294	12.976±1.077
Top50 mean	DNA	**0.970 ± 0.013**	0.966 ± 0.015	0.968 ± 0.016	0.967 ± 0.015	0.967 ± 0.016
	RNA14	**0.927 ± 0.045**	0.897 ± 0.053	0.880 ± 0.059	0.860 ± 0.064	0.922 ± 0.041
	RNA50	**0.768 ± 0.048**	0.753 ± 0.050	0.730 ± 0.050	0.774 ± 0.042	0.754 ± 0.054
	Protein	**14.239±2.206**	11.943±1.006	11.449±1.236	11.563±1.277	12.570±1.041
Top100 mean	DNA	**0.953 ± 0.022**	0.944 ± 0.027	0.950 ± 0.024	0.947 ± 0.026	0.948 ± 0.025
	RNA14	**0.899 ± 0.049**	0.869 ± 0.057	0.847 ± 0.063	0.835 ± 0.064	0.898 ± 0.044
	RNA50	0.737 ± 0.055	0.716 ± 0.060	0.695 ± 0.061	**0.743 ± 0.051**	0.727 ± 0.060
	Protein	**14.020±2.204**	11.754±1.002	11.241±1.213	11.363±1.267	12.358±1.027

Table 3: Different algorithm combinations with EA in the DNA, RNA, and protein design tasks, the mean value comparison of the top 10, 50, and 100 biological sequences.

other predictors across diverse sequence design tasks. This affirms LGFEN's exceptional performance and its suitability for varied design challenges. The experiment that does not use LGFEN but uses CNN as a predictor to compare with baselines like Ren et al. (2022) can be viewed in Appendix B.3, the results show that LGFEN significantly enhances the efficiency of biological sequences. Appendix A.1 contains the implementation specifics of LGFEN.

Effect of Reinforcement Learning

The preceding experiments establish the superiority of the proposed ERLBioSeq algorithm. To assess the efficiency of RL in biological sequence design relative to other models, we conducted additional ablation studies. These studies involved substituting the RL module with alternative modules to ascertain the benefits of RL. We explored the integration of alternative comparative models within the evolutionary process, encompassing Bo, Cmaes, Cbas, and Gflownet. We also evaluated these models on tasks involving DNA, RNA, and protein design tasks. The results of diverse model architectures are presented in Table 3. The evaluation scores represent the average performance scores of the top 10, 50, and 100 sequences generated by distinct model structures. Our approach largely outperforms all other methods, substantiating the exceptional performance resulting from the fusion of EA and RL.

Conclusion and Future Work

In this paper, we introduce a novel biological sequence design approach, termed ERLBioSeq, which does not rely on prior knowledge. Our approach combines the EA and RL, harnessing their respective strengths. Subsequently, we devise a biological scoring predictor LGFEN within the evolutionary process by integrating both local and global information of biological sequences. To assess the efficacy of ERLBioSeq, we compare it with state-of-the-art methods. The evaluation covering DNA, RNA, and protein design shows ERLBioSeq's superiority over alternative state-of-the-art methods. Concurrently, it is established that LGFEN significantly enhances the efficiency of biological sequences. Furthermore, we assessed the impact of various models on the EA. The findings indicate that RL provides the most substantial contribution to the EA. In the future, we intend to undertake targeted optimizations for more specific domains.

Acknowledgments

The work is supported by the National Natural Science Foundation of China (62072376), Guangdong Basic and Applied Basic Research Foundation (2022A1515010144), Innovation Capability Support Program of Shaanxi (2022KJXX-75).

References

Angermueller, C.; Dohan, D.; Belanger, D.; Deshpande, R.; Murphy, K.; and Colwell, L. 2019. Model-based reinforcement learning for biological sequence design. In *International conference on learning representations*.

Arnold, F. H. 1998. Design by directed evolution. *Accounts of chemical research*, 31(3): 125–131.

Auger, A.; and Hansen, N. 2005. Performance evaluation of an advanced local search evolutionary algorithm. In *2005 IEEE congress on evolutionary computation*, volume 2, 1777–1784. IEEE.

Barrera, L. A.; Vedenko, A.; Kurland, J. V.; Rogers, J. M.; Gisselbrecht, S. S.; Rossin, E. J.; Woodard, J.; Mariani, L.; Kock, K. H.; Inukai, S.; et al. 2016. Survey of variation in human transcription factors reveals prevalent DNA binding changes. *Science*, 351(6280): 1450–1454.

Bennett, N. R.; Coventry, B.; Goreshnik, I.; Huang, B.; Allen, A.; Vafeados, D.; Peng, Y. P.; Dauparas, J.; Baek, M.; Stewart, L.; et al. 2023. Improving de novo protein binder design with deep learning. *Nature Communications*, 14(1): 2625.

Brookes, D.; Park, H.; and Listgarten, J. 2019. Conditioning by adaptive sampling for robust design. In *International conference on machine learning*, 773–782. PMLR.

Chaudhury, S.; Lyskov, S.; and Gray, J. J. 2010. PyRosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta. *Bioinformatics*, 26(5): 689–691.

Dama, A. C.; Kim, K. S.; Leyva, D. M.; Lunkes, A. P.; Schmid, N. S.; Jijakli, K.; and Jensen, P. A. 2023. BacterAI maps microbial metabolism without prior knowledge. *Nature Microbiology*, 1–8.

DeLuca, C. I.; Davies, P. L.; Ye, Q.; and Jia, Z. 1998. The effects of steric mutations on the structure of type III antifreeze protein and its interaction with ice. *Journal of molecular biology*, 275(3): 515–525.

Gupta, A.; and Zou, J. 2019. Feedback GAN for DNA optimizes protein functions. *Nature Machine Intelligence*, 1(2): 105–111.

Hansen, N. 2006. The CMA evolution strategy: a comparing review. *Towards a new evolutionary computation: Advances in the estimation of distribution algorithms*, 75–102.

Ho, Q.-T.; Le, N. Q. K.; and Ou, Y.-Y. 2022. m CNN-ETC: identifying electron transporters and their functional families by using multiple windows scanning techniques in convolutional neural networks with evolutionary information of protein sequences. *Briefings in Bioinformatics*, 23(1): bbab352.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Jain, M.; Bengio, E.; Hernandez-Garcia, A.; Rector-Brooks, J.; Dossou, B. F.; Ekbote, C. A.; Fu, J.; Zhang, T.; Kilgour, M.; Zhang, D.; et al. 2022. Biological sequence design with gflownets. In *International Conference on Machine Learning*, 9786–9801. PMLR.

Kuhlman, B.; Dantas, G.; Ireton, G. C.; Varani, G.; Stoddard, B. L.; and Baker, D. 2003. Design of a novel globular protein fold with atomic-level accuracy. *science*, 302(5649): 1364–1368.

Li, C.; Zhang, H.; Ding, P.; Yang, S.; and Bai, Y. 2023. Deep feature extraction in lifetime prognostics of lithium-ion batteries: Advances, challenges and perspectives. *Renewable and Sustainable Energy Reviews*, 184: 113576.

Lorenz, R.; Bernhart, S. H.; Höner zu Siederdissen, C.; Tafer, H.; Flamm, C.; Stadler, P. F.; and Hofacker, I. L. 2011. ViennaRNA Package 2.0. *Algorithms for molecular biology*, 6: 1–14.

Močkus, J. 1975. On Bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference: Novosibirsk, July 1–7, 1974*, 400–404. Springer.

Mourtada, R.; Herce, H. D.; Yin, D. J.; Moroco, J. A.; Wales, T. E.; Engen, J. R.; and Walensky, L. D. 2019. Design of stapled antimicrobial peptides that are stable, nontoxic and kill antibiotic-resistant bacteria in mice. *Nature biotechnology*, 37(10): 1186–1197.

Ren, Z.; Li, J.; Ding, F.; Zhou, Y.; Ma, J.; and Peng, J. 2022. Proximal exploration for model-guided protein sequence design. In *International Conference on Machine Learning*, 18520–18536. PMLR.

Repecka, D.; Jauniskis, V.; Karpus, L.; Rembeza, E.; Rokaitis, I.; Zrimec, J.; Poviloniene, S.; Laurynenas, A.; Viknander, S.; Abuajwa, W.; et al. 2021. Expanding functional protein sequence spaces using generative adversarial networks. *Nature Machine Intelligence*, 3(4): 324–333.

Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. 2020. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Silva, D.-A.; Yu, S.; Ulge, U. Y.; Spangler, J. B.; Jude, K. M.; Labão-Almeida, C.; Ali, L. R.; Quijano-Rubio, A.; Ruterbusch, M.; Leung, I.; et al. 2019. De novo design of potent and selective mimics of IL-2 and IL-15. *Nature*, 565(7738): 186–191.

Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; et al. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419): 1140–1144.

Silver, D.; Singh, S.; Precup, D.; and Sutton, R. S. 2021. Reward is enough. *Artificial Intelligence*, 299: 103535.

Sinai, S.; Wang, R.; Whatley, A.; Slocum, S.; Locane, E.; and Kelsic, E. D. 2020. AdaLead: A simple and robust

adaptive greedy search algorithm for sequence design. *arXiv preprint arXiv:2010.02141*.

Song, Z.; and Li, L. 2023. Importance Weighted Expectation-Maximization for Protein Sequence Design. *arXiv preprint arXiv:2305.00386*.

Storcheus, D.; Rostamizadeh, A.; and Kumar, S. 2015. A survey of modern questions and challenges in feature extraction. In *Feature Extraction: Modern Questions and Challenges*, 1–18. PMLR.

Tinberg, C. E.; Khare, S. D.; Dou, J.; Doyle, L.; Nelson, J. W.; Schena, A.; Jankowski, W.; Kalodimos, C. G.; Johnsson, K.; Stoddard, B. L.; et al. 2013. Computational design of ligand-binding proteins with high affinity and selectivity. *Nature*, 501(7466): 212–216.

Torres, M. D.; Cao, J.; Franco, O. L.; Lu, T. K.; and de la Fuente-Nunez, C. 2021. Synthetic biology and computer-based frameworks for antimicrobial peptide discovery. *ACS nano*, 15(2): 2143–2164.

Trabucco, B.; Geng, X.; Kumar, A.; and Levine, S. 2022. Design-bench: Benchmarks for data-driven offline model-based optimization. In *International Conference on Machine Learning*, 21658–21676. PMLR.

Wang, Y.; Wang, H.; Wei, L.; Li, S.; Liu, L.; and Wang, X. 2020. Synthetic promoter design in Escherichia coli based on a deep generative network. *Nucleic Acids Research*, 48(12): 6403–6412.

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. 2022. Controlling gene expression with deep generative design of regulatory DNA. *Nature communications*, 13(1): 5099.