An Examination of “Designing Biological Sequences without Prior Knowledge Using Evolutionary Reinforcement Learning”

Caleb Alexander
Alex Limoco
Harry Reblando
Ryan Van Witzenburg

August 10, 2025

1 Introduction

This work will serve as an examination and explanation of the paper "Designing Biological Sequences without Prior Knowledge Using Evolutionary Reinforcement Learning" [1].

An active field of study in biology is that of biological sequences, the study of RNA, DNA, and amino acid sequences. However, designing biological sequences is often a lengthy, inefficient, and costly process, particularly when it relies on prior knowledge of the properties of DNA, RNA, and proteins. These adverse factors stem from the vast and complex search space inherent in biological sequence problems, where the sheer number of possible design sequences makes computational determination prohibitive without prior knowledge. The traditional method of studying sequence design are so-called "wet-lab" studies. Which, as the name implies, involve actually creating the sequence in a lab and testing it's properties. Such experiments tend to be resource-intensive, as well as time-consuming. The typical solution to this issue is to design a way of creating and testing a biological sequence virtually. However, existing machine learning models heavily depend on prior biological facts which can limit their overall ability to generalize novel tasks.

Despite the challenges, researchers still desire to study biological sequences. The utilization of biological sequence designs stretches across multiple disciplines such as medicine, biotechnology, and agriculture. Therefore, its potential to make significantly accelerated cross-functional discoveries while reducing the costs of experiments can open new avenues in many fields. For a quick example, we can see potential advancements already taking root within synthetic biology, as enzymes for biofuels are under investigation for breakthroughs [2][3].

Furthermore, the motivations for optimizing biological sequence design in this manner are clear: To encourage innovation. The field is in a deadlock of inefficiencies such as high expenses and slow experimental procedures, but that all could change with a revolutionary reinforcement learning approach. By leveraging these advancements in AI, researchers intend to automate and optimize the biological sequence design process which may open the floodgates for new discoveries.

Above all, the improvements within biological sequence design have various real-world applications. First off, in healthcare, it can revolutionize drug discovery by optimizing sequence search methods. Secondly, in biotechnology, it may advance our knowledge of biofuel which can aid in the reduction of our use of fossil fuels. Finally, in agriculture, it may benefit our understanding of genetically modified (GMO) foods which can help eliminate food deserts both nationally and internationally.

In this paper we will detail the previous methods researchers have used to study biological sequences. Then we will introduce and explain the novel approach used in [1].

2 Current State of Sequence Design

As discussed in the previous section, traditional methods of sequence design involve wet-lab testing. This method was recognizes as expensive and inefficient for years; as such, alternatives were created. One of the most common ways of avoiding wet-lab testing has been through the use of evolutionary algorithms (EAs). These algorithms have been the gold standard approach in sequence design [4], and below we offer an explanation of EAs.

2.1 Evolutionary Algorithms

Evolutionary algorithms aim to reproduce the process of evolution in a set of virtual individuals. EAs have use in many different areas but for our purposes we will discuss their impact as related to biological sequence design. An *individual* in our case will be a particular biological sequence. An evolutionary algorithm will generally proceed as follows:

1. A random set of individuals (sequences) is generated, and generally referred to as a generation.
2. These individuals are evaluated based on a *fitness function*.
3. The *fittest* of the generation are chosen as the *parents* of the next generation.

When a generation is created, the created sequences can be entirely random or randomly chosen within a set of acceptable values. How random or not random the first generation is will often depend on how much prior knowledge exists about the particular biological sequence we desire. For example, if we desire to produce a gene which induces trait X in an organism and we know that any sequence which ends in ACG will certainly **not** induce this trait, then we can eliminate such possibilities from our space of possible individuals. In this case, our starting generation will be chosen randomly from the space of possible gene sequences, not including sequences which end in ACG. In some cases the difference between this space and the space of all possible sequences could be quite large. If we consider the space of all 3-letter sequences, we would be removing 1 of 12 possible sequences. In the case of 10-letter sequences we remove 4^7 sequences of our total 4^{10}. This can be a significant difference which could result in a computational speed up.

Regarding the fitness function, this can often be the most complicated part of an evolutionary algorithm. A fitness function, sometimes referred to as the fitness landscape, is a function which receives a particular sequence of n-length and returns a score of the sequence's *fitness*. In this case, the fitness of a sequence is most likely to be the likelihood that the sequence will invoke a desired trait in an organism.

Generally, when discussing EAs we assume that we already have a suitable fitness function. However, it is worth noting that the creation of a fitness function can be one of, if not the most difficult parts of our entire process. Indeed, this is another area where researchers are limited by the amount of prior knowledge on biological sequences. If we lack the required knowledge to design a sufficient fitness function then our ability to create viable sequences will be limited.

When selecting the fittest sequences of a generation we rank each sequence of the generation by their fitness and keep the top-k (k can often be 10 or 50). These k chosen sequences will then be the *parents* of the next. This means that the sequences will be spliced and mutated such that the resulting sequences are sufficiently different from the original generation. This process will then be repeated until sequences are generated which have reach a desired threshold of fitness.

As discussed in [4], this process can often be described as a maximization optimization problem where we start with a random point (our starting generation) and take steps towards a maximum. In our case, we can think of the starting generation as giving us some idea of what "direction" is best according to fitness. From there we create new generations based on that best direction and add in some randomness to test that we truly have the optimal direction.

Since the introduction of EAs, there have been many attempts to optimize them. Some of these attempts even involved machine learning, as discussed in [5]. However, a novel approach is the integration of reinforcement learning

(RL) as described in [1].

2.2 Reinforcement Learning

As mentioned in the introduction, the limiting factor for most sequence design tasks is the prior knowledge needed. This issue causes most methods of machine learning to be, at the very least, inneficient at the task of sequence design. However, there is a relatively new form of machine learning which requires comparatively less prior knowledge. This form is of course reinforcement learning.

Reinforcement (RL) is a type of machine learning that is derived from behavioral psychology- there is an agent that interacts with an environment taking actions that maximize the amount of reward received. Reinforcement learning differs from other types of machine learning in the way that it does not rely heavily on labeled datasets and does not require extensive prior domain knowledge. RL is optimal for finding solutions when there is little to no prior knowledge and solutions come from exploration and decision making.

In the context of ERLBioSeq, RL provides the ability to bypass the need for prior knowledge that is usually obtained through the extensive wet lab testing that is traditionally done to obtain data on biological sequences. In this context, the RL framework consists of the following elements:

- **Agent:** The RL agent represents the sequence generator. It operates within an extensive design space, making incremental adjustments to sequences.

- **Environment:** The environment includes the rules of sequence formation, constraints on design, and the evaluation process (fitness predictor).

- **Action Space:** The actions in this situation are the modifications to the biological sequences, such as altering the nucleotide or amino acid that is represented in a certain position.

- **Reward Function:** The reward function evaluates the fitness score of sequence (the effectiveness), assigning a higher reward for designs that are more likely to exhibit the desired biological traits (determined by the fitness score).

The RL starts by having the agent randomly generate biological sequences and then receive feedback from the fitness predictor. Over many iterations, through trial and error, the agent will develop a better policy for mapping the sequences to actions. This will lead to the gradual improvement of the quality of the generated biological sequence variants. This approach allows for an exploration of the biological sequence space without requiring the extensive prior knowledge that this environment lacks.

One of the major challenges of the RL is the rareness of meaningful rewards. Sequences with high fitness scores may be scarce within the immense search space of the biological sequence which makes it difficult for the agent to learn effective policies quickly. To address this, evolutionary algorithms are applied to improve learning efficiency.

In [1], RL is used to increase diversity and avoid local optima that commonly occur in evolutionary algorithms. By utilizing RL, the algorithm not only generates biological sequences that have higher fitness scores but also explores new regions in the sequence space, leading to greater overall robustness and innovation in the sequence generation process. This dual approach of harnessing the innovation and learning of RL and the evolutionary selection of the evolutionary algorithms form the basis of the ERLBioSeq algorithm.

3 Proposed Algorithm

The algorithm proposed in [1], named ERLBioSeq, uses EAs and RL in a combines framework such that the whole becomes better than the parts. The workflow can be seen below in figure 1.

The above process can be described in the following steps:

1. Reinforcement learning is used to develop a generation of individuals. (See B in figure 1.)

2. The generation is spliced and mutated.

3. Each individual in the generation has its fitness predicted (See C in figure 1.)

4. The top-k individuals are selected from the generation according to their fitness.

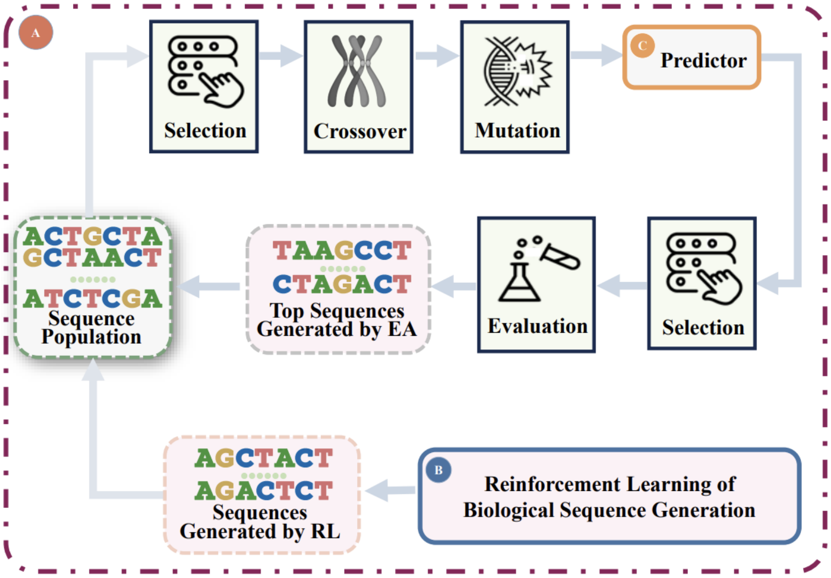

Figure 1: Diagram taken from [1]

(a) Concurrently, reinforcement learning is being used to develop a completely new and separate generation.

5. The top-k sequences and the new generation developed by the RL algorithms are chosen as the next generation and the steps repeat starting from step 2.

The reinforcement learning in this context works similarly to the RL described in previous sections. In this case however, our reward function is set up such that an agent can only maximize its reward by designing sequences which have a higher predicted fitness. The curious reader may wonder why step (4a) is needed. In the section on EAs we describe how the evolutionary process can be described as a maximization process. One un-optimal outcome of any maximization process is the outcome of finding a local maximum rather than the global maximum. If step (4a) was not included, the algorithm would work very quickly, but it is very likely to get stuck in a local maximum since every new generation is directly based on the previous generation. By introducing new unrelated sequences every generation we can mitigate this issue somewhat. At the very least, we decrease the likelihood that our algorithm will be stuck in a severely un-optimal local maximum.

3.1 Contributions

The first and most obvious contribution of the proposed algorithm is the novel combination of RL and EAs. This approach allows for a method of sequence design which quickly grows in accuracy while requiring very little prior knowledge. Another contribution of the paper is the fitness predictor which they designed (See C in figure 1).

Similar to most recent attempts at sequence design, the authors of [1] used convolutional neural network (CNN) as their chosen method of fitness predictor. However, unlike others, they used optimization methods to dramatically increase the efficiency of their predictor. Figure 2 shows the design of said predictor. To describe the details of the predictor mathematically is outside the scope of this paper. However, the important part is that previous efforts did not include the "One-Hot Encoding" part. This encoding allows for the vectors given to the CNN to be compressed and therefore more efficient. Note that this encoding has been constructed with careful attention to including both local and global information preservation. To understand what this means consider a sequence such as

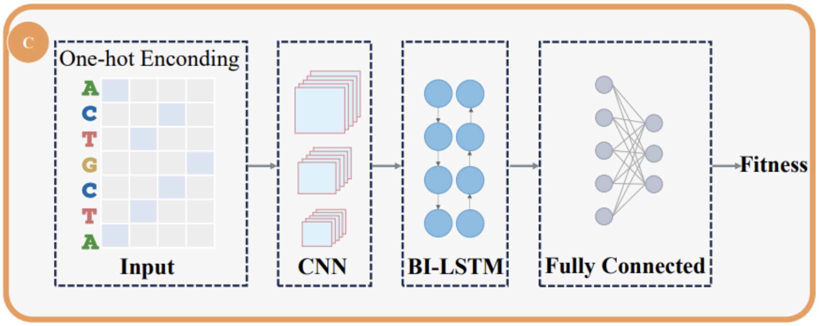

Figure 2: Diagram taken from [1]

$$ACTGGGA$$

and suppose that we know that any gene sequence which includes the string $ACTGG$ induces trait X and any sequence which includes the string GG **and** ends in the letter A induces trait Y. Notice that if we consider the G which is located in the middle of our sequence; it is a part of the subsequence $ACTGG$ which causes trait X **but** it is also causing trait Y since it is a part of GG and our sequence ends in A. In this case, the first part could be considered local information (what are the immediate letters surrounding the current letter), and the second part could be global information (what is true of the sequence as a whole). The predictor designed in [1] takes all of this information into account. This is especially important in biological sequence design where many subsequences, separated far apart, can contribute to the overall traits induced by the sequence as a whole.

4 Results

The ERLBioSeq showed significant improvements in generating biological sequences without prior knowledge. The results presented in [1] demonstrated that using reinforcement learning (RL) with evolutionary algorithms (EAs) led to biological sequence generation that is more efficient and effective compared to traditional and competing approaches.

4.1 Performance Metrics

The overall performance of ERLBioSeq was evaluated on many key metrics.

- **Fitness Improvement:** The generated biological sequences' fitness scores were consistently higher compared to those using other EA algorithms or other conventional methods.

- **Efficiency:** ERLBioSeq required fewer iterations to reach higher fitness scores, meaning there was a major improvement in computational efficiency.

- **Diversity:** The introduction of RL into the framework enabled the exploration of diverse and novel regions in the sequence space, avoiding being trapped in a local optima.

4.2 Comparison to Baseline Methods

Throughout the study, the ERLBioSeq framework was benchmarked against existing sequence design methods to assess its effectiveness. These existing methods included ML models and EAs that stood alone and relied on prior knowledge. The benchmarking results displayed that

- Higher average fitness scores were achieved by the ERLBioSeq framework compared to the prior knowledge EAs
- The ERLBioSeq framework was able to generalize to novel tasks effectively unlike traditional, domain-specific data ML approaches

4.3 Ablation Studies

The contribution of each mechanism in the ERLBioSeq algorithm was measured through ablation studies:

- Removing the RL mechanism reduced the diversity of the biological sequences and slowed the convergence, showing the RL's role in exploration.
- using a less optimized predictor or removing the fitness predictor significantly decreased the overall performance, highlighting its importance.

4.4 Limitations

While there were positive results that highlighted the benefits, two key limitations were identified:

- **Computational Cost:** The use of both the RL and EA, along with the fitness predictor, increased computational demands.
- **Reward Design:** The reward functions in the RL algorithm required careful calibration in order to balance exploitation and exploration, causing concern for scalability in other domains.

Overall, the results confirmed the effectiveness of the ERLBioSeq algorithm in addressing the hurdles of biological sequence design while also offering opportunities for more optimizations and wider applications. Its innovative and unique approach paves the way for advancements in other industries such as drug discovery, and biotechnology.

5 Discussion

The ERLBioSeq framework has made leaps and bounds for computational sequence designing. With that being said, there are still improvements we see that may enhance performance and increase its impact and applicability. The glaring limitation falls back on the restriction to high-quality data. So, if the datasets used in training were to expand in size and diversity of data, the predictive accuracy would increase and the model would become more generalized. Moreover, we would aim to refine the criteria for the sequence space in order to encourage the framework to focus on more decisive regions of DNA, RNA, and protein sequence spaces. This would, in turn, increase efficiency while reducing the computational demand. Similarly, we would improve the accuracy in tandem with the precision of the initial sequence generation which could provide a better foundation for optimization with higher fitness scores.

With that, another avenue to improve this study is to enhance the fitness predictor. The virtual sequences that are generated by ERLBioSeq are translated into desired traits much more efficiently if more detailed biological insights are used, i.e. a higher fitness predictor. Additionally, advanced RL algorithms like CrossQ, which, integrates batch normalization to enhance sample efficiency, could reduce computational demands.

Beyond improving ERLBioSeq, the techniques used throughout this study may be used to address other problems. For example, in agriculture, RL and EAs may benefit our understanding of genetically modified (GMO) foods which can help eliminate food deserts both nationally and internationally. Also, the framework of the study may lead to revolutionization in the healthcare industry with optimized sequence search methods.

Finally, we have found that simplifying this framework could lead to it being more accessible and efficient. Tightly integrating RL and EA alongside a finely nurtured fitness predictor could bring unification in the field while maintaining the needed high performance. We see future work as improving virtual sequences by seamlessly translating them to real world traits. This translation would result in a trustworthy environment where computationally generated results reliably produce the desired traits. Implementing these framework simplifications can become a powerful tool and may become a cornerstone for AI-driven biological sequence designing and beyond.

References

[1] Xi Zeng, Xiaotian Hao, Hongyao Tang, Zhentao Tang, Shaoqing Jiao, Dazhi Lu, and Jiajie Peng. Designing biological sequences without prior knowledge using evolutionary reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(1):383–391, Mar. 2024.

[2] Daniel-Adriano Silva, Shawn Yu, Umut Y. Ulge, Jamie B. Spangler, Kevin M. Jude, Carlos Labão-Almeida, Lestat R. Ali, Alfredo Quijano-Rubio, Mikel Ruterbusch, Isabel Leung, Tamara Biary, Stephanie J. Crowley, Enrique Marcos, Carl D. Walkey, Brian D. Weitzner, Fátima Pardo-Avila, Javier Castellanos, Lauren Carter, Lance Stewart, Stanley R. Riddell, Marion Pepper, Gonçalo J. L. Bernardes, Michael Dougan, K. Christopher Garcia, and David Baker. De novo design of potent and selective mimics of il-2 and il-15. *Nature*, 565(7738):186–191, 2019.

[3] Nathaniel R. Bennett, Brian Coventry, Inna Goreshnik, Buwei Huang, Aza Allen, Dionne Vafeados, Ying Po Peng, Justas Dauparas, Minkyung Baek, Lance Stewart, Frank DiMaio, Steven De Munck, Savvas N. Savvides, and David Baker. Improving de novo protein binder design with deep learning. *Nature Communications*, 14(1):2625, 2023.

[4] Frances H. Arnold. Design by directed evolution. *Accounts of Chemical Research*, 31(3):125–131, 03 1998.

[5] Sam Sinai, Richard Wang, Alexander Whatley, Stewart Slocum, Elina Locane, and Eric D. Kelsic. Adalead: A simple and robust adaptive greedy search algorithm for sequence design. *CoRR*, abs/2010.02141, 2020.