Imperial College London

MSc Individual Project - Background Report

Imperial College London

Department of Computing

Offline Model-Based Optimisation for Controllable Biological Sequence Design Using QD and Foundation Models

Author:
Alona Rimon

Supervisor:
Dr. Antoine Cully

Second Supervisors:
Mr. Maxime Allard
Ms. Hannah Janmohamed

June 4, 2025

Contents

Chapter 1

Introduction

1.1 Motivation

Offline model-based optimisation is the setting where, given an offline dataset of solutions (e.g. designs) and their scores based on an unknown objective, the optimisation process aims to output the best possible solution without querying that objective function. This problem framework is highly relevant in scenarios where evaluating the objective (from now on referred to as the oracle) is expensive and time-consuming, such as in materials discovery, robot design, neural architecture search, and molecule or protein design [1, 2].

A particularly suitable application is the design of biological sequences. Sequences such as RNA, DNA, and proteins are central to the development of vaccines, therapeutics, and synthetic biology. Since in vivo and in vitro experiments are costly and slow, the in-silico design stage is essential for generating candidate sequences before wet-lab evaluation. Generating high-quality novel sequences computationally can significantly accelerate discovery without relying on exhaustive and expensive experiments.

In practice, users of in-silico sequence generation systems are rarely interested in arbitrarily realistic sequences; rather, they seek sequences that satisfy specific user-defined constraints (e.g., stable mRNA, high translation efficiency, low immunogenicity). This can be viewed as a controllable sequence design under the offline model-based optimisation setting.

Foundation models such as ProGen [3], Helix-mRNA [4], and Evo [5] have emerged as natural tools in this space, building on the success of language modelling and exploiting the analogy between natural language and biological sequences. These models are often pre-trained on massive unlabelled datasets and encode useful biological priors. While their use as sequence generators is possible, their application to controllable design remains largely open. Since they were trained in a non-aligned, generic fashion, using them effectively for controllable tasks, especially ones that are defined by an unknown oracle, is a non-trivial challenge.

Another important consideration is the need to generate multiple diverse candidates. In practice, many in-silico designs fail in wet-lab experiments due to unforeseen structural, regulatory, or functional issues. Therefore, it is crucial to not only optimise for the desired properties (fitness) but also ensure diversity among the generated sequences—maximising the chance of success by covering different modes of the sequence-function landscape. Moreover, allowing users to influence the space of diversity can support a more interactive and informative design process. Rather than being used solely for final optimisation, design algorithms can aid early-stage ideation and abstraction, helping users understand the range of possible solutions and refine their objectives [6].

We believe that Quality-Diversity (QD) algorithms [7], which aim to illuminate the search space by identifying diverse yet high-performing solutions, offer a promising approach to this challenge. QD methods like MAP-Elites [8] have demonstrated strong results in domains where the objective is complex, multimodal, and partially observed.

Combining foundation models with QD optimisation is a relatively unexplored direction in biological sequence design. If successful, this approach could lead to more reliable and controllable discovery pipelines. It may also provide a general framework for offline optimisation problems in other domains where data is limited and diversity is essential, leading to faster, cheaper, and more reliable candidate discovery.

1.2 Objectives

The objectives of this project are:

1. Develop and implement a Quality-Diversity optimisation framework for biological sequence generation, tailored for offline model-based optimisation. The approach is inspired by the method introduced by Jérémie Dona et al. [9].
2. Leverage foundation models (e.g., Helix-mRNA [4], Evo2 [5], HyenaDNA [10], Caduceus [11]) to support various roles in the design process, including guiding sequence exploration, acting as surrogate scoring models, deriving behavioural descriptors, or some combination thereof.
3. Evaluate the proposed framework in terms of both performance and diversity, and analyse the respective contributions of foundation models and Quality-Diversity algorithms within the offline model-based optimisation paradigm—both in general and specifically for biological sequence design.
4. Investigate and address the limitations of standard distance metrics (e.g., Euclidean distance in continuous spaces and Levenshtein distance in sequence spaces) in capturing biologically meaningful variation. In biological sequence design, even minor nucleotide changes can result in significant structural and functional shifts. To this end, this project aims to develop and evaluate improved diversity and novelty metrics that are more aligned with biological function and structure, enabling better interpretability and insight into sequence variation.
5. Conduct comparative analyses between the proposed method and established offline sequence design benchmarks to rigorously assess its relative strengths, weaknesses, and generalization capability.

1.3 Challenges

Several core challenges complicate offline biological sequence design:

Firstly, the solution space is high-dimensional, vast, and discrete. For example, a 50-nucleotide RNA sequence has 4^{50} possible configurations. Yet, only a narrow manifold within this space corresponds to valid biological sequences, and an even smaller fraction exhibits useful functionality for a given task [1].

Secondly, offline optimisation is constrained by limited, task-specific data. Offline model-based optimisation lacks the ability to query new sequences, which makes model generalisation particularly challenging. Learned scoring models often become unreliable when evaluating novel, out-of-distribution (OOD) sequences. Common issues in offline design optimisation include overestimation of fitness on OOD samples and reward hacking, both contributing significantly to the sim-to-real gap between computational predictions and wet-lab outcomes [1].

Thirdly, the data distribution is often heavy-tailed and non-representative of the true design space. Many datasets contain only a sparse sampling of high-fitness designs, meaning that performant sequences often lie on a thin manifold that is poorly covered by training data [1]. This makes it difficult to learn accurate models and increases the risk of proposing unrealistic sequences.

Fourth, the objective functions in biological sequence design are often highly sensitive. Small changes in sequence space—such as a single nucleotide substitution—can lead to dramatic shifts in fitness. Consequently, proximity in sequence space does not imply similar objective values, which hinders gradient-based optimisation strategies and requires models to be especially cautious when exploring new regions of the space. Moreover, certain biological properties remain inherently challenging or impossible to accurately predict in silico.

Lastly, diversity among candidate solutions is critical. Different sequences may perform well under varying experimental conditions, so maximising the chance of downstream success requires generating candidates that span diverse modes of the fitness landscape [9, 12]. However, defining an appropriate space for diversification and choosing a distance metric—e.g., sequence similarity, structural differences, or functional properties—is non-trivial and depends on the downstream use case.

1.4 Problem Formulation

We formally define the task of controllable biological sequence generation as an *offline model-based Quality-Diversity optimisation problem.* The problem is defined as follows.

Given:

- A finite alphabet $\mathcal{A}$ (e.g., nucleotides or amino acids),
- A sequence space $\mathcal{X} = \mathcal{A}^L$, consisting of all sequences of length L,
- An unknown true objective function $F_f : \mathcal{X} \to \mathbb{R}$ (the oracle function), capturing the desired biological property,
- An offline labelled dataset $D = \{(x_i, y_i)\}_{i=1}^N$ with $x_i \in \mathcal{X}$ and $y_i = F_f(x_i)$,
- A known behavioural descriptor function $F_b : \mathcal{X} \to \mathbb{R}^d$, mapping sequences to behaviour descriptors (e.g., motifs, annotations, predicted properties),
- A discretised behavioural space $\mathcal{B}$, obtained by partitioning the continuous descriptor space into K bins (cells).

The goal is to return a batch of K sequences $\{x_i\}_{i=1}^K$ that together approximate a diverse set of high-performing solutions across the behavioural space. Specifically, these sequences should:

1. Achieve high scores under F_f,
2. Exhibit high diversity in the behavioural descriptor space $\{F_b(x_i)\}_{i=1}^K$, ideally covering a significant portion of $\mathcal{B}$.

Formally, the optimisation target is:

$$\forall b \in \mathcal{B}, \quad x_b^* = \arg\max_{x \in \mathcal{X}} F_f(x) \quad \text{subject to} \quad F_b(x) \in b.$$

Since the true optimum x_b^* for each bin b is unknown, the returned batch of K solutions serves as an approximation, aiming to include at least one high-performing sequence per behaviour bin in $\mathcal{B}$.

Offline setting. Crucially, we operate without access to online queries of the oracle function F_f, relying solely on the offline dataset D to build surrogate models and guide optimisation.

This formulation inherently emphasises the discovery of a diverse set of high-performing solutions under offline constraints. Details on surrogate model implementation, offline optimisation frameworks, the use of foundation models, and specific evaluation strategies are further discussed in Chapters 2,3.

Chapter 2

Literature Review

2.1 Offline Model-Based Optimisation

2.1.1 Definition

In offline model-based optimisation, the goal is for an algorithm A to output a batch B of K candidate solutions, given access only to a static dataset $D = \{(x_i, f(x_i))\}$ of previously evaluated designs and their corresponding scores. The true objective function f is unknown and cannot be queried during optimisation. Instead, the algorithm must rely on models learned from the offline data to propose high-performing solutions.

This setting is especially relevant in domains where evaluating the objective (also called the oracle) is costly or time-consuming, such as materials discovery, robotic design, neural architecture search, or biological sequence design. [1, 2]

Note that multi-objective optimisation (MOO) involves different problem formulations and evaluation metrics. In this work, we focus exclusively on the single-objective setting.

2.1.2 General Approaches

In offline model-based optimisation, methods differ substantially in both their choice of surrogate model and their approach to candidate generation. The surrogate model serves as an approximation of the unknown objective function (oracle) using the offline dataset. Since the oracle cannot be queried during optimisation, the surrogate becomes the primary guide for selecting promising candidates.

A major challenge in this context is the risk of out-of-distribution (OOD) overestimation: a surrogate may assign high scores to regions of the design space far from the training data, where its predictions are unreliable. Dominant recent works, therefore, focus on making surrogates more robust, especially near the boundaries of the dataset's support [1, 13, 2].

In addition to surrogate modelling, the candidate generation process is another critical design axis. One popular approach, for example, will be to use gradient ascent in the input space steered by surrogate model predictions. [13, 14]. The challenge is to balance *novelty*, *diversity*, and *trustworthiness*: proposed candidates must not only score well according to the surrogate but also remain plausible under the true (unknown) objective. This has motivated a view of offline MBO as a form of **controllable generation**, in which the goal is to model a conditional distribution $p(x \mid c)$, with c representing the target property value as measured (or approximated) by the oracle. From Bayes' rule, we get that modelling $p(x \mid c)$ is proportional to the modelling of $p(x) \cdot p(c \mid x)$. While there are methods that use inverse generation approach - directly capturing $p(x \mid c)$ (e.g. conditional VAE [15], conditional GANs [16], and more), we focus here on methods following the approach named forward modelling - steering generative model prior $p(x)$ using a surrogate model modelling $p(c \mid x)$ [17, 2].

Surrogate Models. Surrogates vary from simple fully connected neural networks to more complex ensembles and uncertainty-aware models. A particularly influential class is Conservative Objective Models (COMs), which aim to underestimate rather than overestimate the oracle score in regions of uncertainty [13]. COMs apply adversarial regularisation to penalise optimistic predictions far from training data, helping mitigate reward hacking and leading to more reliable opti-

misation steps. Other approaches use Bayesian neural networks [18], or dropout-based uncertainty estimation [19].

Generative Models Guided by Surrogates. Some methods combine generative models with surrogate scores to guide design. VAEs map sequences to a latent space where gradient ascent can be applied, followed by decoding to generate candidates [20]. Diffusion models are increasingly used for realistic, high-quality generation, often conditioned on surrogate scores [21]. Autoregressive models may also be trained with reward gradients or within a reinforcement learning framework. For example, model-based RL methods like DyNA-PPO leverage surrogate models for efficient reward-guided sequence generation in low-round, large-batch biological design settings [22].

Ultimately, the generative process must navigate a tension between exploiting known high-performing regions of the design space and exploring diverse, novel sequences that may generalise beyond the training data.

2.1.3 Biological Sequence Domain

Offline model-based optimisation has become increasingly relevant in the domain of biological sequence design, where each sequence—whether a protein, DNA, or RNA variant—must be optimised for functional properties such as expression level, stability, or binding affinity. In this setting, wet-lab experiments are expensive and slow, which makes the one-shot, offline paradigm a practical necessity.

A growing number of benchmarks and methods have been proposed to study this problem. The **Design-Bench** suite [1] provides a standardised set of offline biological optimisation tasks including GFP fluorescence prediction, 5'-UTR translation efficiency, and transcription factor binding. These datasets enable fair comparison of offline algorithms using realistic biological sequences and oracle models as black-box evaluators.

Several offline optimisation methods have been proposed or adapted for these biological domains. DyNA-PPO [22] proposes a model-based RL algorithm with an offline training loop that fits surrogate models on labelled sequences and uses them for policy updates.

In addition to reinforcement learning, methods such as GFlowNets [12] have been applied to biological sequence generation, emphasising the production of diverse, high-scoring candidates. These approaches are well-suited to biological design tasks where multiple functional variants may be viable, and where covering different regions of sequence space is desirable due to uncertainty in in-silico evaluation.

A recent and relevant direction is the use of surrogate models that incorporate pre-trained biological language models. Chen et al. [14] propose a surrogate that adds a linear prediction head on top of a frozen pre-trained LM, and applies deep linearisation to make bidirectional learning tractable. This allows the surrogate to benefit from rich biophysical features learned from millions of sequences, overcoming the representational limitations of previous NTK-based approaches.

This approach is relevant to our work, as it demonstrates how domain-aware pre-trained models can be leveraged in offline MBO to improve both fidelity and robustness of the surrogate model, while enabling more informed and biologically grounded optimisation.

Across these works, a recurring challenge is balancing exploration and exploitation in high-dimensional, discrete, and poorly understood sequence spaces.

2.1.4 Evaluation Metrics

The evaluation of offline MBO solutions is still an evolving field. While early work mainly focused on performance (or usefulness) [1], it's now standard to also consider additional metrics such as diversity and novelty [12, 9, 23], and more recently, measures like stability [24] and coverage [23].

Let $A(D) = B$ be the output of an algorithm A given an offline dataset D, where B is a batch of K proposed solutions. Several key metrics are used to evaluate this batch [2]:

Usefulness

Usefulness (or performance) is measured using an oracle model f that estimates the ground truth objective. In many tasks (e.g., 5'UTR design), an exact oracle is unavailable. In such cases, the standard approach is the one proposed by Trabucco et al. [1]: (i) train a high-quality surrogate oracle on the data, (ii) sample a portion (for example the lower performing half) of this data as the offline dataset D, and (iii) evaluate the batch B at the end using the oracle f.

The oracle must not be queried during optimisation—this would be equivalent to accessing the true reward function, which we aim to avoid. In real-world problems like biological sequence design, querying the oracle is analogous to performing costly wet-lab experiments.

The mean and maximum oracle score over the batch are used for evaluation:

$$\text{mean}(B) = \frac{1}{|B|} \sum_{(x_i, y_i) \in B} y_i, \quad \max(B) = \max_{(x_i, y_i) \in B} y_i$$

where each score is normalised using the min and max values from the full dataset:

$$y_i = \frac{f(x_i) - f_{\min}}{f_{\max} - f_{\min}}.$$

[1]

Diversity

Beyond performance, we usually want diverse solutions. A diverse set is more likely to include candidates with different functional modes, which can be important for robustness— increasing the chance for one of them to succeed in the real-world experiments. [12]

Diversity is typically defined as the average pairwise distance between elements in B:

$$\text{Diversity}(B) = \frac{1}{K(K-1)} \sum_{i \neq j} d(x_i, x_j)$$

where d is a distance function.

Coverage Coverage is another way to look at diversity. Instead of measuring pairwise distance, it measures how well the proposed batch spreads across the design space. A recently preposed metric is the L1 coverage [23]:

$$L1C = \frac{1}{d} \sum_{k=1}^{d} \max_{i \neq j} |x_{ik} - x_{jk}|$$

assuming a continuous representation of the solutions in d dimensions.

Novelty

Novelty is essential in any design task. If an algorithm proposes solutions that already exist in the offline dataset, those aren't really useful. To ensure novelty, we compare each proposed solution to a reference set (usually the offline training data):

$$\text{Novelty}(B) = \frac{1}{K} \sum_{i=1}^{K} \min_{j=1,\ldots,N_{\text{ref}}} d(x_i, x_j^{\text{ref}})$$

where d is again a distance function.

Distance $d(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ For continuous domains, Euclidean distance is often used; for sequences, edit distance (e.g., Levenshtein) is a standard choice [2].

Stability

Stability was recently proposed by Qian et al. [24] as a way to capture how consistent an optimisation method is. It measures performance throughout the optimisation process, compared to the best solution in the offline data. This is especially useful in offline MBO, where it's hard to determine when to stop the optimisation process. A stable optimiser is one that consistently finds better-than-dataset solutions early on and continues to improve.

Summary

Together, these metrics paint a fuller picture of an offline MBO algorithm's behaviour: Usefulness captures absolute performance; diversity, coverage, and novelty quantify exploration; and stability assesses optimisation robustness.

2.2 Quality-Diversity Optimisation

2.2.1 Background

Quality-Diversity optimisation is a framework that aims to generate a collection of diverse, high-performing solutions rather than a single optimum. It is particularly well-suited for problems in which success depends not only on maximising a surrogate score but also on capturing a wide range of functional behaviours. The flagship algorithm in this space, MAP-Elites [8], discretises a user-defined behaviour descriptor space and fills it with solutions that maximise performance within each cell—an approach known as *illumination* [7].

2.2.2 Relevant works in Offline MBO and Biology Design

While QD methods have shown promise in scientific domains such as robotics and materials discovery—for instance, in crystal structure prediction, where Janmohamed et al.[25] applied Multi-Objective QD to uncover diverse, stable structures with desirable physical properties—their application to biological sequence design, particularly in offline settings, remains largely unexplored. In these contexts, exploration is as important as exploitation, as generating diverse batches of candidate sequences can increase the likelihood of discovering viable solutions. This is especially critical in the **one-shot setting**, where only a single batch can be experimentally validated. To the best of our knowledge, the only work applying QD to offline biological sequence design is that of Donà et al.[9].

Donà et al. introduce a QD framework specifically tailored to one-shot biological sequence optimisation. Their method builds on MAP-Elites and explicitly addresses the constraints of the offline setting. It integrates conservative ensemble-based fitness estimation (COMs) [13] to mitigate overestimation in out-of-distribution regions, defines behaviour descriptors based on pairwise similarity to sequences in the offline dataset, explores the sequence space using randomised switch (flip) mutations, and applies Centroidal Voronoi Tessellation (CVT) for downsampling and final batch selection.

Although designed for offline, one-shot optimisation, the method relies on random mutation operators and an MLP ensemble-based surrogate, which may be limited in their ability to explore the vast and high-dimensional space of biological sequences. In essence, the approach does not fully capture the rich grammar, structure, and contextual dependencies inherent in biological sequences—factors that are often critical for identifying diverse and functionally relevant candidates.

In our work, we build on this foundation by investigating how domain-specific prior knowledge can be incorporated into offline MBO to more effectively guide exploration and improve sample efficiency.

Another work by Boige et al. [26] proposes ME-GIDE, a MAP-Elites variant that uses gradients of the objective and descriptor functions to guide mutations in discrete spaces. This work focuses on accelerating QD in discrete spaces using gradient-informed mutation, but it does not consider the offline optimisation setting where the scoring function is learned from a limited dataset and cannot be queried. As such, it targets a different problem- online search with unlimited oracle access, and is not directly applicable to offline, one-shot biological sequence design.

Another relevant approach is the Surrogate-Assisted Illumination (SAIL) algorithm [27], which combines MAP-Elites with a Gaussian Process surrogate to reduce the number of expensive evaluations. SAIL builds an acquisition map by maximising an uncertainty-aware acquisition function (e.g., UCB) within each cell, guiding sampling toward informative regions of the behaviour space.

While SAIL was originally proposed for online settings with expensive physical evaluations, its core ideas—combining QD with active surrogate modelling—are particularly relevant to the offline MBO setting. Unlike Donà et al. [9], who focus on conservative learning and descriptor design, SAIL provides a general strategy for sample-efficient illumination via uncertainty-aware acquisition. However, SAIL relies on a continuous input space and classical GPs, which may not scale well to high-dimensional discrete sequence design or incorporate structure-aware biological priors. Additionally, SAIL was not originally designed for the strict evaluation budget of the one-shot regime.

Overall, while these approaches represent a promising direction for biological sequence design using QD, further work is needed to improve sample efficiency, incorporate richer biological priors, and better handle the limitations of offline, uncertain scoring environments.

2.2.3 Quality-Diversity Metrics

Quality-Diversity optimisation is typically evaluated using the following standard metrics:

1. **QD score** – the sum of fitness values across all filled cells in the repertoire.
2. **Max/Mean fitness** – measures of performance across all solutions in the repertoire.
3. **Behaviour Descriptor (BD) coverage** – the number of filled niches, indicating the diversity of explored solutions.

These metrics are computed based on the fitness model, usually a surrogate in the offline MBO setting. However, since surrogate models can suffer from overestimation, especially in out-of-distribution regions, high QD scores or fitness values during optimisation may not reflect the true quality of the solutions. This limitation applies specifically to fitness-dependent metrics such as QD score and max/mean fitness, but not to BD coverage, which only reflects the diversity of explored solutions and is independent of the predicted fitness values.

Therefore, in offline MBO, it is essential to assess final solution quality using oracle-based evaluations, as outlined in Section 2.1.4. A cautionary example appears in ME-GIDE [26], where the solutions with the highest predicted fitness (according to the surrogate) were clearly flawed when visualised. This is acceptable in their case because they assume an online setting with access to the true fitness function.

In contrast, our offline setting assumes no such access, making it crucial to distinguish between surrogate-based metrics during optimisation and oracle-based evaluations for measuring real-world usefulness.

2.3 Foundation Models in Bio-Design

Foundation models — large neural networks trained on comprehensive biological datasets — have emerged as powerful tools for modelling and designing biological sequences. Analogous to language models in NLP, these models capture statistical and functional patterns across genomic or protein sequence space, enabling applications such as property prediction, mutation effect estimation, and sequence generation [4, 5, 10, 28, 17, 29].

Despite their power, foundation models are not plug-and-play solutions for controllable design tasks, nevertheless those with offline sparse data, as in our settings. Most are trained to model natural sequence distributions, not to optimise for downstream properties. As such, their outputs may be diverse but not functionally optimal.

Nonetheless, foundation models offer rich representations and inductive biases that can be leveraged for controllable generation and surrogate learning. In this work, we aspire to leverage them both for search space navigation and exploration as generators, and as when properly aligned with design objectives— they have the potential to significantly improve biological sequence design.

2.4 Aligning Foundation Models

Controllable biological sequence design with foundation models can be viewed through the broader lens of *alignment*—the task of guiding large pretrained models to follow user intent. In natural language processing, this has led to techniques such as instruction tuning (RLHF) [30], aimed at steering foundation models toward producing outputs that are both high-quality and aligned with user-defined goals.

Similar principles are beginning to appear in biological sequence design. For example, ProteinRL [31] fine-tunes a pretrained protein language model using reinforcement learning to optimise sequence outputs for properties such as charge or solubility.

While this represents a form of alignment between generative models and user-defined objectives, it operates in an online setting and assumes full access to the reward function during training. These reward functions are relatively simple and engineered—such as net charge or sequence similarity—rather than complex black-box oracles. The approach does not model the reward function explicitly, nor does it account for the constraints of respecting a fixed, offline dataset that defines the user's design intent. Nonetheless, it reflects the broader idea of aligning generative models to downstream biological objectives.

Other recent works also aim to steer generative models toward biological constraints, including diffusion-based models conditioned on structure or function [32], and autoregressive models such as ProGen2 [3] conditioned on functional annotations.

These efforts point to a growing convergence between alignment strategies in NLP and controlled sequence generation in biology. Yet, extending such techniques to the offline setting—where user intent is expressed implicitly through limited, static datasets—remains an open and challenging problem.

Chapter 3

Technical Achievements And Experimentation

3.1 Experimental Setup

We build on the implementation and methods introduced in [1] (https://github.com/rail-berkeley/design-bench) as a foundation for defining the task and the oracle.

3.1.1 5'UTR Task

The 5'UTR task involves designing mRNA sequences of length 50 that achieve maximal ribosome load. To this end, we use the 5'UTR dataset provided in Design-Bench. Following the approach of Trabucco et al., we also adopt the filtering procedure from Sample et al. [33], selecting the top 280,000 sequences with the highest total read counts. This yields a dataset of mRNA sequences of length 50 paired with their corresponding ribosome load values. In line with [9], we apply min-max normalisation to the ribosome load values to scale them into the range $[0, 1]$.

3.1.2 Oracle

Following [1] and [9], we use an approximate oracle based on a ResNet architecture, as the available data does not cover all possible sequences. We adopt the same architecture as used in the Design-Bench repository: an embedding layer followed by a two-layer residual convolutional neural network, which is further followed by a self-attention layer and a linear projection. The input data is preprocessed using z-normalisation to have zero mean and unit variance.

To verify that our oracle is comparably reliable to those used in [9] and [1], we evaluate its performance on held-out validation data and compare it to the reported metrics. As shown in Table 3.1, our oracle achieves a Spearman correlation that falls between the results reported in the aforementioned works.

	Spearman Correlation
One-shot reported	0.887
Design-Bench reported	0.8617
Ours	0.8707

Table 3.1: Comparison of oracle Spearman correlation. Our oracle shows comparable performance to previously reported results.

3.1.3 Offline Dataset

The trained oracle is used to re-label the full dataset with predicted fitness scores.

To mimic real-world constraints—where high-performing sequences are rare—we construct the offline dataset by sub-sampling a third of the full 5'UTR dataset with probability proportional to $\exp(-f(x))$, where $f(x)$ is the oracle-predicted fitness of sequence x. This sampling scheme intentionally emphasises lower-performing regions of the search space, reflecting realistic scenarios in which only suboptimal designs are initially available.

This offline dataset is used as the input for the downstream optimisation procedure.

3.1.4 Objectives

All evaluation metrics are computed using the oracle model as a proxy for true fitness. Following the metrics defined in Section 2.1.4, given an output batch B of K solutions, we report:

- **Performance:** Mean and maximum oracle-predicted fitness, where $f(x)$ is the oracle function estimating the true objective:

$$\text{mean}(B) = \frac{1}{|B|} \sum_{(x_i, y_i) \in B} y_i, \quad \max(B) = \max_{(x_i, y_i) \in B} y_i, \quad y_i = \frac{f(x_i) - f_{\min}}{f_{\max} - f_{\min}}$$

- **Diversity / Novelty:**
  - *Diversity:* Average pairwise distance between proposed solutions:

$$\text{Diversity}(B) = \frac{1}{K(K-1)} \sum_{i \neq j} d(x_i, x_j)$$

  - *Novelty:* Average distance to nearest neighbour in the reference set:

$$\text{Novelty}(B) = \frac{1}{K} \sum_{i=1}^{K} \min_{j=1,\ldots,N_{\text{ref}}} d(x_i, x_j^{\text{ref}})$$

  - **Distance functions used** (d)**:** In classical Offline Model-Based Optimization (MBO) algorithms, it is common to measure distances in the primary sequence space[2]. However, sequence space distances often do not correlate well with functional or structural similarity: small sequence changes may lead to drastic changes in folding or function, while large sequence edits might have little to no biological effect. Therefore, we aim to explore and compare different distance spaces to better capture biologically meaningful diversity. Specifically, we consider the following distance functions:
    - Levenshtein edit distance (in sequence space),
    - Levenshtein edit distance of secondary structures (computed using ViennaRNA [34]),
    - Cosine distance in the oracle's internal 120-dimensional embedding space.

3.1.5 Constrains

The optimisation algorithm does not have access to the oracle, neither during the search process nor for hyperparameter tuning.

3.2 Method

The main components of our method are presented below. For additional technical details and a complete list of hyperparameters, refer to Appendix A.

3.2.1 Optimisation with CVT-MAP-Elites

For implementing the optimisation algorithm, we build on the codebase provided in [35], where Bradley et al. integrate language models with Quality-Diversity (QD). We adapt the implementation to match our experimental requirements.

To discretize the behavioural descriptor space into cells, we use Centroidal Voronoi Tessellation (CVT)[36], where the centroids are determined by clustering the behavioural descriptors of the offline dataset. This ensures that the partitioning of the space is adapted to the distribution of known biologically relevant sequences. We then implement the MAP-Elites algorithm over this discretised space: at each iteration, a batch of solutions is sampled from the currently filled cells in the repertoire, mutated using one of the mutation models described below, and reinserted into the repertoire based on their surrogate-predicted fitness and behavioural descriptor.

3.2.2 Grid Initialisation and Behaviour Descriptors

Following the approach of [9], the repertoire is initialised with a batch of sequences sampled from the offline dataset. To ensure a fair comparison across runs, all optimisation procedures start from the same initial population.

Behaviour descriptors are defined as 3-dimensional vectors representing the normalised frequencies of nucleotides (A, C, G) within each sequence. Although there are four nucleotide types, the frequency of the fourth (U) can be inferred as one minus the sum of the others, making three dimensions sufficient. Min-max normalisation is applied based on statistics computed from the offline dataset.

3.2.3 Surrogate Fitness Models

We employ two surrogate models trained on the offline dataset: a fine-tuned version of the Helix-mRNA foundation model and an ensemble of multi-layer perceptrons (MLPs) trained using the Conservative Objective Models (COMs) strategy.

Model	MSE	Spearman	Pearson
Fine-tuned Helix-mRNA	0.0037	0.856	0.861
COMs Ensemble	0.0084	0.767	0.823

Table 3.2: Performance of surrogate fitness models on held-out validation data

The **COMs Ensemble** model architecture follows the same setup used in [9], with each MLP consisting of two hidden layers of 2048 units and ReLU activations. Training follows the COMs strategy [13] to ensure conservative behaviour on out-of-distribution inputs. Each MLP in the ensemble is trained independently on the offline dataset using a loss function that includes both standard supervised regression and adversarial penalisations on overconfident predictions for perturbed (potentially out-of-distribution) inputs. At inference time, predictions are aggregated using the conservative estimate:

$$\text{fitness}(x) = \mu(x) - \beta \cdot \sigma(x),$$

where μ and σ are the mean and standard deviation across the ensemble predictions, and $\beta \in \mathbb{R}$ is a tunable conservativeness factor.

The **Helix-mRNA** model is the foundation model publicly released by Helical-AI [4], available at `https://www.helical-ai.com/`. We fine-tune it on the offline 5'UTR dataset by appending a linear regression head and unfreezing the final two layers during training to allow moderate adaptation.

3.2.4 Mutation Strategies

We evaluate two mutation operators for generating new candidate sequences during MAP-Elites iterations. Both strategies operate on a randomly selected contiguous substring of fixed length (5 nucleotides, i.e., 10% of the sequence length):

- **Helix-mRNA generator**: A generative mutation strategy that replaces the selected substring with a new segment sampled from the pre-trained Helix-mRNA foundation model.
- **Random mutator**: A baseline strategy that applies uniform random substitutions over the selected 5-nucleotide region. Each position in the substring is replaced independently with a randomly sampled nucleotide.

3.2.5 Downsampling to Final Batch

The full MAP-Elites run maintains a large repertoire (e.g., 2000 cells) to explore a broad portion of the search space. To extract a final batch of $K = 128$ solutions, and using the approach of [9], we perform a second CVT-MAP-Elites pass using a reduced grid of 128 cells. All solutions from the original repertoire are reinserted into this reduced grid, and the elite from each new cell is selected. The centroids for the reduced CVT grid are computed using k-means clustering over the

behavioural descriptors of the existing solutions. We also report the top 128 sequences selected by the oracle-predicted score.

3.3 Results

In this section, we compare four optimization configurations using the MAP-Elites algorithm. All experiments used nucleotide frequencies as the behavioural descriptor, and each configuration was run with five different random seeds. The same initial batch of solutions was used across all configurations for each seed. The configurations tested are:

- **Random mutator + Ensemble COMs fitness** (baseline configuration from [9]),
- **Random mutator + Helix-mRNA fitness**,
- **Helix-mRNA mutator + Ensemble COMs fitness**,
- **Helix-mRNA mutator + Helix-mRNA fitness**.

3.3.1 Fitness, Diversity, and Novelty Statistics

We report the mean and maximum fitness values with respect to the oracle for each configuration, alongside diversity and novelty scores computed using secondary structure distances. Results are presented for two evaluation settings: (i) the downsampled set of K solutions obtained via Centroidal Voronoi Tessellation, which reflects the final output of the algorithm, and (ii) the top-K solutions ranked by oracle score across the full repertoire, which serves as an upper bound for performance and is not accessible during optimization.

Configuration	Mean Fitness	Max Fitness	Diversity	Novelty
Random Mutator + COMs Fitness	0.9152	1.0067	21.4302	3.5953
Random Mutator + Helix Fitness	0.8757	0.9698	21.4872	**4.0586**
Helix Mutator + COMs Fitness	**0.9313**	**1.0579**	**21.7808**	3.9328
Helix Mutator + Helix Fitness	0.8782	0.9981	20.9783	3.8078

Table 3.3: Mean and maximum oracle-predicted fitness, as well as diversity and novelty with respect to secondary structure, computed for the downsampled-K solutions in each configuration.

Configuration	Mean Fitness	Max Fitness	Diversity	Novelty
Random Mutator + COMs Fitness	0.9863	1.0247	19.3824	2.6500
Random Mutator + Helix Fitness	0.9466	0.9793	18.5128	2.9023
Helix Mutator + COMs Fitness	**1.0076**	**1.0627**	**21.0080**	3.0359
Helix Mutator + Helix Fitness	0.9645	1.0257	19.7705	**3.2906**

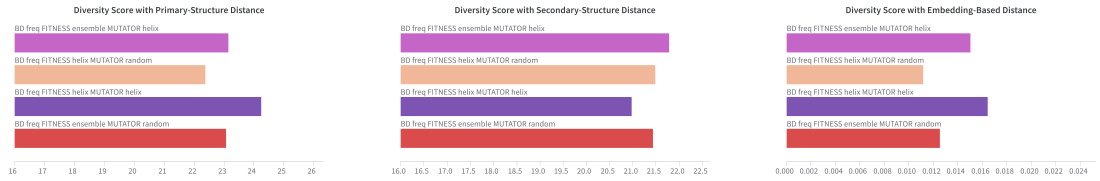
Table 3.4: Mean and maximum oracle-predicted fitness, as well as diversity and novelty with respect to secondary structure, computed for the top-K oracle-scoring solutions in each configuration.

3.3.2 Diversity and Novelty

To assess the behavioural spread of the generated sequences, we evaluate each configuration on both diversity and novelty throughout the optimisation process using three different distance metrics:

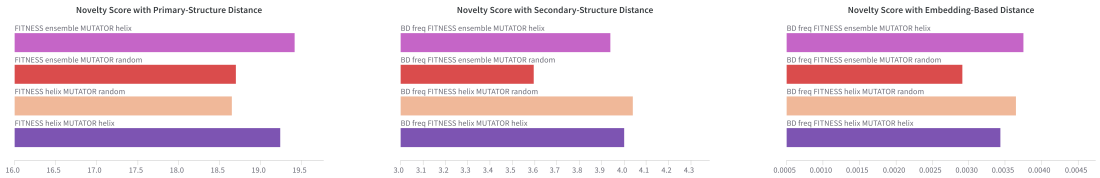
- Levenshtein distance on the primary (nucleotide) sequence,
- Levenshtein distance on the predicted secondary structure (secondary structure computed via ViennaRNA[34]),
- Cosine distance in the oracle's embedding space.

Figure 3.1 shows the diversity scores for each of the three distance measures, all computed over the final batch of the downsampled repertoire. Figure 3.2 shows the corresponding novelty scores.



(a) Primary sequence (b) Secondary structure (c) Embedding space (Cosine)

Figure 3.1: Diversity of the final batch for each configuration, measured using three distance metrics.



(a) Primary sequence (b) Secondary structure (c) Embedding space (Cosine)

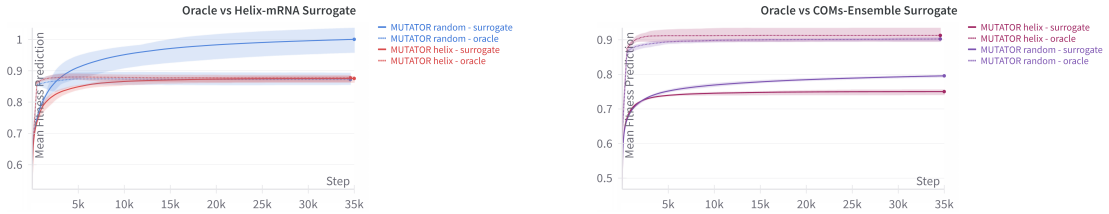
Figure 3.2: Novelty of the final batch for each configuration, measured using three distance metrics.

3.3.3 Fitness Prediction Error

A central component of our optimisation framework is the use of surrogate models to estimate the fitness of candidate sequences instead of costly or unavailable oracle evaluations. To analyse the behaviour and reliability of the surrogate models during the optimisation process, we compare the predicted fitness scores from the surrogate with the true oracle scores.

Figure 3.3 presents these comparisons for two configurations: one using the fine-tuned Helix-mRNA model and another using the COMs ensemble. In each subplot, the solid line represents the surrogate predictions, while the dashed line corresponds to oracle scores.

It is important to note that oracle values are not accessible during actual optimisation and are shown here only for post hoc analysis to better understand the dynamics and limitations of each surrogate model.



(a) Helix-mRNA as surrogate (b) COMs ensemble as surrogate

Figure 3.3: Comparison of surrogate-predicted fitness (solid lines) and oracle scores (dashed lines) during the optimisation process. Although Oracle values are used here for analysis, they are not available to the algorithm during the optimisation process.

3.4 Results Analysis and Discussion

Usefulness Analysis As shown in Table 3.3, the most effective configuration in terms of usefulness combines Helix-mRNA as the mutator with the COMs ensemble as the fitness predictor. Compared to the random mutator, Helix-guided mutations lead to higher-quality sequences, likely

due to their proximity to the training distribution, which improves surrogate reliability and mitigates reward hacking — the exploitation of model inaccuracies on out-of-distribution (OOD) samples.

This suggests that Helix-mRNA serves as a structured mutator, producing biologically plausible sequences that are more faithfully scored by COMs. As a result, search is guided away from unrealistic regions and toward regions where surrogate estimates are more trustworthy.

Surrogate vs. Oracle Fitness Analysis To better understand these dynamics, Figure 3.3 compares the progression of mean fitness as estimated by both the surrogate models and the oracle, across different mutator–scorer combinations. Several consistent patterns reinforce the explanation above.

First, both Helix-mRNA and COMs surrogates consistently assign higher fitness to sequences from the random mutator, while the oracle ranks Helix-mutated sequences higher. This mismatch suggests that random mutations more often produce out-of-distribution samples that the surrogate overestimates — a classic case of reward hacking. In contrast, Helix-guided mutations stay closer to the training distribution, leading to more reliable surrogate evaluations. This is consistent with Table 3.3, where the Random Mutator + Helix Fitness Surrogate configuration yields the most novel but least useful sequences.

Second, oracle scores tend to plateau early, while surrogate scores — especially under random mutation — continue to rise. This may reflect a growing divergence between optimised sequences and the surrogate's support region, with the surrogate maintaining overconfidence even as uncertainty increases. This underscores the risk of unconstrained surrogate-guided optimisation and motivates future work on uncertainty estimation or support-aware sampling. These observations demand further investigation and analysis to better understand the underlying dynamics and to develop more robust optimisation strategies.

Third, Helix-mRNA underperforms COMs as a fitness model across mutators, even though it achieves better performance on the validation set, as shown in 3.2. This may indicate greater vulnerability to OOD exploitation, possibly due to architectural or training differences. In contrast, COMs are explicitly trained to be conservative on unsupported inputs, likely contributing to their robustness. This suggests a promising direction: incorporating conservative principles into Helix-mRNA as a fitness model to reduce reward hacking and improve generalisation.

Top-k vs. Downsampled Set Comparing Table 3.3 (CVT-based downsampling) to Table 3.4 (oracle-based top-k selection), we see that relative performance across configurations remains generally consistent. However, as expected, top-k selection yields higher usefulness but sacrifices diversity and novelty, while CVT provides a more balanced trade-off.

This trade-off highlights an opportunity: using surrogate scores to guide downsampling may enable more nuanced control over this balance. If surrogate predictions correlate well with the oracle, one could support user-defined diversity–fitness trade-offs. To enable this, further investigation is needed into the surrogate–oracle correlation across the solution repertoire.

Distance Measures Analysis Figures 3.1 and 3.2 show inconsistent rankings between runs when using different distance metrics, indicating that these measures capture distinct notions of similarity. This supports the view that conventional diversity metrics, based purely on primary sequence distance, are limited in biological relevance.

In contrast, metrics based on predicted structural or physicochemical properties might reflect functionally meaningful diversity. We propose that benchmarks should explicitly define the behaviour space in which diversity is desired, aligned with the design goal. Depending on the task, this space could include structural motifs, physicochemical attributes, or predicted function profiles.

This aligns with practitioner workflows, where in-silico tools are used iteratively, not as end-to-end solutions [27]. In this context, behaviour-aware diversity may help users explore viable alternatives, especially when paired with human judgement.

Chapter 4

Project Plan

4.1 Project Plan

This section outlines the month-by-month plan for the remaining stages of the project, leading up to the final submission on September 10th.

June

- **Model analysis and refinement:**
  - Investigate the surrogate model's reliability and robustness.
  - Analyse and explain discrepancies where the oracle remains flat while the surrogate rises.
- **Pipeline extension:**
  - Integrate at least one additional DNA-based task into the framework.
  - Explore and test uncertainty estimation strategies for the surrogate model.
  - Attempt to incorporate likelihood estimates from the Helix-mRNA generator into the optimisation loop.
  - Experiment with a probabilistic modelling approach:

    $$p(\mathbf{x} \mid \mathbf{y}_c) \propto p(\mathbf{x}) \cdot p(\mathbf{y}_c \mid \mathbf{x})$$

    where $p(\mathbf{x})$ is estimated by the generator and $p(\mathbf{y}_c \mid \mathbf{x})$ by the surrogate predictor, enabling more principled fitness estimation under conditional constraints.

July

- **Behaviour descriptor (BD) investigation:**
  - Experiment with alternative BD definitions and evaluate their effect on diversity and performance.
- **Benchmark experiments:**
  - Compare the proposed method against the following methods:
    - GRAD (gradient ascent on surrogate [1])
    - CbAS (conditioned sampling [15])
    - CMA-ES (evolution strategy as used in [1])
    - REINFORCE (policy gradient for sequence design as in [1])
    - COMs (conservative objective modelling [13])
    - QD-One-Shot (Map-Elite-based optimisation with random exploration guided by COMs fitness model [9])
  - Evaluate on metrics such as performance, diversity, and novelty using UTR and other relevant tasks.

August

- **Generator alignment:**
  - Investigate in-loop generator (mutator) adaptation by aligning its sampling distribution with successful sequences found in previous iterations.
  - If time permits, explore light fine-tuning to bias generation towards promising regions.
- **Dataset extension and final experimentation:**
  - Add another task (preferably DNA or protein-based) to demonstrate generalizability.
- **Report writing:**
  - Begin drafting full sections of the report, incorporating finalised results, figures, and comparative analysis.

September

- **Final submission:**
  - Final edits to the report.
  - Submit project on or before **September 10th**.

4.2 Further Directions

Although not currently included in the core project timeline, the following research directions present promising avenues for future exploration:

1. **Offline multi-objective optimisation for biological sequence design.** Extending the current framework to the multi-objective setting may allow for simultaneous optimisation of trade-offs. This could leverage Pareto-based methods techniques.
2. **Generalisation to non-biological sequence design tasks.** The methods developed in this project could be adapted for offline optimisation in other domains, such as superconducting materials or ant morphology design. These tasks often present similar challenges in terms of sparse, high-quality data and the need for diversity-aware search.
3. **Hybrid offline-online active learning loop.** One compelling extension is integrating a limited-query online phase into the offline pipeline. Specifically, sequences that induce high epistemic uncertainty in the surrogate could be selected for oracle querying, thereby maximising information gain.

Appendix A

Additional Technical Information and Hyperparameters

This appendix provides additional implementation details and the specific hyperparameters used throughout this work.

MAP-Elites. The algorithm was executed for 35,000 iterations. In each iteration, a batch of 128 genomes was sampled from the repertoire. Every genome in the batch was independently mutated and then reinserted into the repertoire. The mutation operator consisted of replacing a continuous chunk of five nucleotides within the sequence. The final reported set of optimized candidates, denoted by K, was a batch of size 128. Novelty scores were computed using all the offline data sequences as a reference set.

COMs ensemble fitness model. For the ensemble-based fitness model, we used the same architecture and hyperparameters for the Conservative Objective Models (COMs) as described by Trabucco et al. [13] and Donà et al. [9]. The ensemble comprises four fully connected neural networks, each with two hidden layers of width 2048, followed by a ReLU activation function. The input to each network is a relaxed, continuous approximation of a one-hot encoded sequence. This relaxation is computed using the formula $\tilde{X} = \log(CX + \frac{1.0-C}{K})$, where $C = 0.6$, X is the one-hot encoded input, and $K = 4$ is the number of nucleotides in the mRNA alphabet. Since this transformation introduces a linear dependency among the coordinates of $\tilde{X}$, the final dimension is removed to eliminate redundancy.

Each COM-MLP was trained with mean squared error loss, using the Adam optimizer with a learning rate of 3×10^{-4}, a batch size of 128, and for 50 epochs. We used additional COM-specific hyperparameters: $\alpha_{\text{init}} = 0.1$, $\alpha_{\text{lr}} = 0.01$, an overestimation limit of 2.0, a particle learning rate of 2.0, and 50 particle steps. As described in Section 3.2.3, we compute the final conservative fitness score using the formula $fitness(x) = \mu - \beta \cdot \sigma$, where μ and σ are the mean and standard deviation across ensemble predictions, and β is set to 2.0.

Helix-mRNA fitness model. The Helix-mRNA model, developed by Helical-AI [4] (https://www.helical-ai.com/), was fine-tuned as a fitness model using a regression head added to the pre-trained model. Only the final two layers of the network were unfrozen during fine-tuning. We trained the model on the offline dataset using the mean squared error loss, the AdamW optimizer, a batch size of 128, and a learning rate of 1×10^{-4} for 10 epochs.

Helix-mRNA generator. To repurpose the Helix-mRNA model as a mutator, we exploited its autoregressive nature. During mutation, a continuous chunk of 5 nucleotides in the sequence is randomly selected. If the chosen chunk starts at the first position (index 0), the first nucleotide is sampled randomly. The remainder of the mutated region is then generated autoregressively by feeding the model the preceding nucleotides and selecting the most likely next base at each step.

Finally, to ensure compliance with the offline model-based optimization setting, all hyperparameters were tuned based solely on surrogate model validation scores. No ground-truth oracle queries outside the provided dataset were used. All models were trained exclusively on the offline dataset, with evaluations conducted on a validation set split from this dataset.

Bibliography

[1] Trabucco B, Geng X, Kumar A, Levine S. Design-bench: Benchmarks for data-driven offline model-based optimization. In: International Conference on Machine Learning. PMLR; 2022. p. 21658-76.

[2] Kim M, Gu J, Yuan Y, Yun T, Liu Z, Bengio Y, et al. Offline Model-Based Optimisation: Comprehensive Review. arXiv preprint arXiv:250317286. 2025.

[3] Nijkamp E, Ruffolo JA, Weinstein EN, Naik N, Madani A. Progen2: exploring the boundaries of protein language models. Cell systems. 2023;14(11):968-78.

[4] Wood M, Klop M, Allard M. Helix-mRNA: A Hybrid Foundation Model For Full Sequence mRNA Therapeutics. arXiv preprint arXiv:250213785. 2025.

[5] Brixi G, Durrant MG, Ku J, Poli M, Brockman G, Chang D, et al. Genome modeling and design across all domains of life with Evo 2. BioRxiv. 2025:2025-02.

[6] Bradner E, Iorio F, Davis M, et al. Parameters tell the design story: ideation and abstraction in design optimization. In: Proceedings of the symposium on simulation for architecture & urban design. vol. 26. Citeseer; 2014. p. 1-8.

[7] Chatzilygeroudis K, Cully A, Vassiliades V, Mouret JB. Quality-diversity optimization: a novel branch of stochastic optimization. In: Black Box Optimization, Machine Learning, and No-Free Lunch Theorems. Springer; 2021. p. 109-35.

[8] Mouret JB, Clune J. Illuminating search spaces by mapping elites. arXiv preprint arXiv:150404909. 2015.

[9] Jérémie D, Flajolet A, Marginean A, Cully A, Pierrot T. Quality-diversity for one-shot biological sequence design.

[10] Nguyen E, Poli M, Faizi M, et al. HyenaDNA: Long-Range Genomic Sequence Modeling at Single Nucleotide Resolution. arXiv preprint arXiv:230615794. 2023.

[11] Schiff Y, Kao CH, Gokaslan A, Dao T, Gu A, Kuleshov V. Caduceus: Bi-directional equivariant long-range dna sequence modeling. arXiv preprint arXiv:240303234. 2024.

[12] Jain M, Bengio E, Hernandez-Garcia A, Rector-Brooks J, Dossou BF, Ekbote CA, et al. Biological sequence design with gflownets. In: International Conference on Machine Learning. PMLR; 2022. p. 9786-801.

[13] Trabucco B, Kumar A, Geng X, Levine S. Conservative objective models for effective offline model-based optimization. In: International Conference on Machine Learning. PMLR; 2021. p. 10358-68.

[14] Chen C, Zhang Y, Liu X, Coates M. Bidirectional learning for offline model-based biological sequence design. In: International Conference on Machine Learning. PMLR; 2023. p. 5351-66.

[15] Brookes D, Park H, Listgarten J. Conditioning by adaptive sampling for robust design. In: International conference on machine learning. PMLR; 2019. p. 773-82.

[16] Kumar A, Levine S. Model inversion networks for model-based optimization. Advances in neural information processing systems. 2020;33:5126-37.

[17] Zhu Y, Kong Z, Wu J, Liu W, Han Y, Yin M, et al. Generative AI for Controllable Protein Sequence Design: A Survey. arXiv preprint arXiv:240210516. 2024.

[18] Hernández-Lobato JM, Adams R. Probabilistic backpropagation for scalable learning of bayesian neural networks. In: International conference on machine learning. PMLR; 2015. p. 1861-9.

[19] Gal Y, Ghahramani Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. PMLR; 2016. p. 1050-9.

[20] Gómez-Bombarelli R, Wei JN, Duvenaud D, Hernández-Lobato JM, Sánchez-Lengeling B, Sheberla D, et al. Automatic chemical design using a data-driven continuous representation of molecules. ACS central science. 2018;4(2):268-76.

[21] Chen C, Herpoldt KL, Zhao C, Wang Z, Collins M, Shang S, et al. AffinityFlow: Guided Flows for Antibody Affinity Maturation. arXiv preprint arXiv:250210365. 2025.

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

[23] Yao MS, Gee JC, Bastani O. Diversity By Design: Leveraging Distribution Matching for Offline Model-Based Optimization. arXiv preprint arXiv:250118768. 2025.

[24] Qian H, Zhu Y, Shu X, Liu S, Wen Y, An X, et al. SOO-Bench: Benchmarks for Evaluating the Stability of Offline Black-Box Optimization. In: The Thirteenth International Conference on Learning Representations; 2025. .

[25] Janmohamed H, Wolinska M, Surana S, Pierrot T, Walsh A, Cully A. Multi-Objective Quality-Diversity for Crystal Structure Prediction. In: Proceedings of the Genetic and Evolutionary Computation Conference; 2024. p. 1273-81.

[26] Boige R, Richard G, Dona J, Pierrot T, Cully A. Gradient-informed quality diversity for the illumination of discrete spaces. In: Proceedings of the Genetic and Evolutionary Computation Conference; 2023. p. 119-28.

[27] Gaier A, Asteroth A, Mouret JB. Data-efficient exploration, optimization, and modeling of diverse designs through surrogate-assisted illumination. In: Proceedings of the Genetic and Evolutionary Computation Conference; 2017. p. 99-106.

[28] Li S, Moayedpour S, Li R, Bailey M, Riahi S, Kogler-Anele L, et al. Codonbert: Large language models for mrna design and optimization. bioRxiv. 2023:2023-09.

[29] Consens ME, Dufault C, Wainberg M, Forster D, Karimzadeh M, Goodarzi H, et al. Transformers and genome language models. Nature Machine Intelligence. 2025:1-17.

[30] Ouyang L, Wu J, Jiang X, Almeida D, Wainwright C, Mishkin P, et al. Training language models to follow instructions with human feedback. Advances in neural information processing systems. 2022;35:27730-44.

[31] Sternke M, Karpiak J. ProteinRL: Reinforcement learning with generative protein language models for property-directed sequence design. In: NeurIPS 2023 Generative AI and Biology (GenBio) Workshop; 2023. .

[32] Watson JL, Juergens D, Bennett NR, Trippe BL, Yim J, Eisenach HE, et al. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. BioRxiv. 2022:2022-12.

[33] Sample PJ, Wang B, Reid DW, Presnyak V, McFadyen IJ, Morris DR, et al. Human 5 UTR design and variant effect prediction from a massively parallel translation assay. Nature biotechnology. 2019;37(7):803-9.

[34] Lorenz R, Bernhart SH, Höner zu Siederdissen C, Tafer H, Flamm C, Stadler PF, et al. ViennaRNA Package 2.0. Algorithms for molecular biology. 2011;6:1-14.

[35] Bradley H, Fan H, Carvalho F, Fisher M, Castricato L, reciprocated, et al.. OpenELM; 2023. Available from: https://github.com/CarperAI/OpenELM.

[36] Vassiliades V, Chatzilygeroudis K, Mouret JB. Using centroidal voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm. IEEE Transactions on Evolutionary Computation. 2017;22(4):623-30.