

Optimization of DNA Constructs for Gene Expression Guided by Machine Learning

João Lima¹, Sofia Ferreira², and Miguel Rocha³

¹ School of Engineering, Minho University, Campus de Azurém, 4800-019, Guimarães, Portugal cp@eng.uminho.pt
<http://www.eng.uminho.pt>

² Informatics Department, University of Minho, 4710-057 Braga, Portugal

1 Introduction

Recent advances in artificial intelligence (AI) have transformed biology by enabling machine learning (ML) algorithms to analyze large genomic datasets and uncover intrinsic patterns [6]. While models like AlphaFold have revolutionized protein structure prediction [10], applying ML to DNA remains limited due to regulatory complexity and the scarcity of high-quality experimental data [16], hindering progress in synthetic biology.

Foundation Models (FMs) offer a promising approach, as they are pre-trained on massive datasets and can generalize effectively even with limited experimental input [11]. Evo [16], a recent FM, shows potential in decoding nucleotide sequences at single-nucleotide resolution across multiple biological modalities (DNA, RNA, and proteins). Tools like ART also demonstrate ML applications in strain engineering [23], though they face challenges in modeling complex biological interactions.

This work aims to expand Evo’s capabilities by integrating it with new datasets to predict optimal combinations of regulatory elements—particularly promoters and ribosome binding sites (RBS)—to enhance gene expression in *Escherichia coli*. Optimizing these genetic constructs has direct implications for the biotechnological production of biofuels, vitamins, and pharmaceuticals.

2 State of the Art

Microorganisms have naturally evolved to prioritize survival by directing metabolic resources toward biomass production, a configuration often suboptimal for industrial purposes. Synthetic biology offers a powerful toolkit to reprogram cellular functions and redirect metabolism toward the efficient synthesis of high-value compounds [3,13].

Industrially robust strains frequently lack native biosynthetic pathways for target products. Even when such pathways are introduced, reconfiguring metabolism is often necessary to increase the flux of key precursors toward the desired end-product [13]. This is typically achieved through the introduction of heterologous genes and/or the overexpression of native ones.

Precise control of gene expression is central to this strategy, with promoters—specific DNA sequences that regulate the initiation and strength of transcription. By engineering or selecting promoters with defined properties, researchers can fine-tune when, where, and how strongly genes are expressed [22].

These advances in regulatory control have enabled efficient microbial production of biofuels, pharmaceuticals, and specialty chemicals through targeted genetic manipulations [3,13,18].

Synthetic biology transforms traditional production methods by reprogramming living systems to synthesize commercially valuable compounds. Chen et al. [3] and Lee et al. [13] highlight how systems-level strategies and synthetic biology tools empower microbial platforms to produce biofuels, pharmaceuticals, and specialty chemicals. Notable achievements include yeast engineered for bio-ethanol production with enhanced inhibitor tolerance [5], improved yields of polyhydroxyalkanoates (PHAs) in microbial systems through CRISPR-Cas9 genome editing [27], and increased productivity of food industry compounds enabled by CRISPR-based genetic modifications [18].

Precise gene expression regulation appears as a common thread in these studies. González [22] demonstrates that modular cloning in model organisms yields insights into transgene control, while Lv et al. [14] and Oesterle et al. [17] report that automated DNA assembly and CRISPR-derived tools deliver the fine-tuning needed to optimize metabolic pathways. These findings collectively reinforce that meticulous control of gene expression is foundational to metabolic and biosynthetic engineering for efficient production of compounds with commercial interest.

2.1 Regulatory Elements of Gene Expression

Promoters are ubiquitous genetic elements that drive gene transcription, characterized by two conserved regions approximately 35 and 10 base pairs (bp) upstream of the transcription start site (-35 and -10 regions, respectively) [1]. These elements affect the frequency and location of transcription initiation through interactions with RNA polymerase. Although they have been used to regulate gene expression, native promoters lack continuous regulatory strength and broad regulatory scope [8], limiting their application in synthetic systems requiring precise control. As such, promoter engineering is essential to overcoming these limitations, enabling precise regulation of gene expression in modified organisms.

In bacteria, ribosome binding sites (RBSs) are nucleotide sequences located upstream of the start codon in an mRNA transcript, responsible for recruiting the ribosome during translation initiation. RBSs, like other RNA regulatory sequences, are essential elements for translation control. Consequently, they are often mutated to optimize genetic circuits, metabolic pathways, and recombinant protein expression. The interaction between promoters and RBSs influences global gene expression, creating an additional level of complexity in optimizing synthetic systems and making gene expression prediction a considerable challenge [7,2,19,4].

2.2 Machine Learning Approaches in Synthetic Biology

Ideally, the nucleotide sequence of a promoter would provide sufficient information to accurately predict its transcriptional strength across different biological contexts. In practice, however, promoter behavior is influenced by complex *in vivo* interactions—such as chromatin structure, transcription factor availability, and genomic context—that are difficult to model deterministically. Machine learning (ML) techniques have emerged as powerful tools to address this challenge by leveraging large-scale sequence-activity datasets to uncover hidden regulatory patterns [9].

As comprehensively reviewed by de Jongh et al. [9], ML-based approaches can now successfully construct models predicting gene expression levels from regulatory sequences, forming the cornerstone of algorithms that enable rational design of regulatory regions with specific expression levels. These models employ a range of supervised learning methods, including linear regression and support vector machines for element- or k-mer-based features, as well as deep neural networks capable of learning directly from nucleotide sequences. Iterative mutation algorithms and element selection strategies further leverage trained ML models to navigate the vast sequence design space toward desired expression profiles.

Key successes include the use of deep learning to optimize 5'-UTRs in yeast and the application of these computational methods to design synthetic promoters with tailored strengths. The review highlights that while ML offers a data-driven alternative to traditional biophysical models, challenges remain in interpretability, cross-species applicability, and integration of dynamic regulatory contexts [9].

With the growing availability of large datasets, these approaches are increasingly enabling synthetic biologists to design highly specific genetic building blocks for complex pathways and circuits. However, further work is needed to improve model generalizability and incorporate multi-scale regulatory features to fully realize the potential of ML in promoter design [9].

Previous approaches to predicting promoter strength have used techniques such as Support Vector Machines (SVMs) [15], but they often faced limitations due to the complexity of sequence-function interactions and the scarcity of comprehensive experimental data. These limitations have hindered the development of truly predictive promoter design tools.

Similarly, AI models struggle to generate effective Ribosomal Binding Site (RBS) sequences due to the complex interplay of factors influencing translation initiation rates. Traditional models rely on thermodynamic calculations, which often fail to capture RNA secondary structures and their effects on RBS strength, leading to inaccurate predictions [26]. Additionally, most AI approaches are constrained by short context lengths and cannot fully model long-range interactions within the genome [26]. Unlike previous models, Evo can integrate DNA, RNA, and protein information, enabling it to predict how RBS variations affect gene expression with high accuracy. By leveraging deep signal processing techniques, Evo generates biologically plausible sequences, potentially optimizing RBS and promoter design for synthetic biology applications [25].

2.3 Foundation Models in Genomics

In the field of machine learning applied to biology, Evo is a foundation model designed to capture two fundamental aspects of biology: the multimodality of the central dogma and the multiscale nature of evolution. The central dogma integrates DNA, RNA, and proteins with a unified code and predictable information flow, while evolution unifies the drastically different length scales of biological function represented by molecules, pathways, cells, and organisms [25].

Evo overcomes previous model limitations through a deep signal processing-based architecture, scaled to 7 billion parameters with a context length of 131 kilobases at single-nucleotide resolution. Trained on 2.7 million prokaryotic and phage genomes, Evo demonstrates zero-shot functional prediction across DNA, RNA, and protein modalities, competing with or surpassing domain-specific language models [16]. The model employs the StripedHyena architecture, integrating 29 Hyena layers with 3 Rotary Attention layers, achieving subquadratic complexity that enables efficient processing of long genomic sequences. This hybrid design not only enhances scalability and performance compared to traditional Transformers but also increases biological relevance, making Evo particularly well-suited for genome-scale analysis and generation tasks [16,20].

Evo implements single-nucleotide resolution tokenization, treating each base (A, T, C, G) as an individual token. The system uses UTF-8 encoding to map nucleotides to integer values, which are then converted into dense embeddings via a lookup layer. This approach preserves sensitivity to point mutations and small-scale variations, and includes special tokens for specific generation tasks such as CRISPR-Cas design [16].

This combination of advanced tokenization and architecture allows Evo to capture evolutionary patterns and enable prediction and design at the genomic scale, representing a significant advancement over previous models that focused on a single biological modality or scale. The efficiency in handling long sequences makes it possible to process complete genome sequences of millions of organisms [16].

3 Proposed Methodology

This project aims to expand Evo, a foundational genomic model, by integrating a new dataset and leveraging transfer learning to predict optimal combinations of promoters and ribosome binding sites (RBS) for expression in *E. coli*. The proposed approach combines advanced machine learning techniques with experimental validation, creating a feedback loop for continuous model refinement.

3.1 Data Acquisition and Processing

The first step of the work plan involves identifying a comprehensive and reliable experimental dataset, as data quality and coverage are critical for success of downstream modelling. To ensure robust training and validation, the project will

prioritize well-established, experimentally validated sources that provide broad sequence diversity and standardized expression measurements:

- **Kosuri et al. (2013) Dataset [12]:** This dataset comprises 12,653 experimentally tested promoter-RBS combinations, providing a solid foundation for initial model training. It was chosen due to its broad sequence space coverage and robust gene expression quantification methods.
- **IGEM Registry:** With over 20,000 documented genetic parts, this resource will complement the main dataset with additional regulatory elements and application contexts.
- **Additional Data on Artificial Promoters and Engineered Elements:** To enhance dataset diversity, synthetic elements designed for specific applications will be incorporated.

Data preparation will consist of several key steps. First, filtering will be performed to select well-annotated promoter-RBS combinations with reliable expression data. Following this, expression values will be normalized to enable accurate comparisons across different studies. Additionally, sequences will be annotated with relevant metadata, including information about their source, experimental context and functional roles when available. Finally, the dataset will be divided into training, validation, and test sets to ensure representativeness and robustness in model evaluation.

3.2 Model Implementation

The Evo model will be extended and adapted to incorporate the new dataset, with a particular focus on its ability to interpret concatenated promoter-RBS-gene sequences. Evo was selected as the foundation due to its demonstrated proficiency in modeling long DNA sequences and capturing relevant evolutionary patterns. To ensure seamless integration, the tokenization process will align with Evo’s pre-training structure, preserving its established scheme while maintaining the model’s capacity to identify critical contextual relationships within the sequences. Additionally, various supervised learning algorithms will be implemented and systematically compared to optimize performance:

1. Support Vector Machines (SVMs) using Scikit-learn: Selected for their effectiveness in high-dimensional spaces and ability to handle non-linear relationships through appropriate kernels [21].
2. Feed-forward Neural Networks (FNNs) using PyTorch: Chosen for their flexibility and ability to model complex feature interactions [24].

Additionally advanced neural network architectures, including attention layers and residual connections, will be explored to enhance the model’s predictive capacity for unseen promoter-RBS combinations.

Performance evaluation will be conducted using various metrics, including accuracy, recall, AUROC (Area Under the Receiver Operating Characteristic curve), and the Pearson correlation coefficient. These metrics were selected due to their relevance in assessing the model’s ability to predict gene expression accurately in biological contexts. Accuracy and recall will provide insights into the model’s classification performance, while AUROC will measure its ability

to distinguish between positive and negative classes. The Pearson correlation coefficient will assess the degree of linear relationship between predicted and observed gene expression levels, offering a quantitative measure of prediction consistence.

3.3 Experimental Validation

The predictions of the model will be experimentally evaluated using a well-established benchmark coding gene sequence that is known for its robust expression in *Escherichia coli* cells, in collaboration with the Systems and Synthetic Biology Lab at ITQB-NOVA. This experimental validation is crucial to assess the practical applicability and reliability of the model's predictions.

4 Work Plan

The project will be structured in the following phases:

1. Data Collection and Preparation

The initial phase involves compiling the dataset from Kosuri et al. (2013) [12] and integrating data from the iGEM Registry along with additional data on artificial promoters and engineered elements. This will be followed by annotation and filtering of data to ensure quality. Development of normalization and annotation protocols will complete this phase.

2. Adaptation of the Evo Model

This phase focuses on extending the model to interpret concatenated sequences and adjusting tokenization structure for compatibility. We will implement the neural network architecture for transfer learning and conduct preliminary tests with data subsets to verify functionality.

3. Implementation of Machine Learning Algorithms

The implementation phase includes configuring Support Vector Machines (SVMs) using Scikit-learn and implementing Feedforward Neural Networks (FNNs) using PyTorch. We will experiment with advanced neural network architectures and perform hyperparameter optimization with comparative evaluation of results.

4. Experimental Validation (collaboration with ITQB-NOVA)

For validation, we will design fluorescence experiments to test predictions, construct and test selected promoter-RBS combinations, quantify gene expression, and analyze results. The model will be refined based on these experimental outcomes.

5. Final Analysis and Documentation

The concluding phase includes comprehensive statistical analysis of results and comparison with existing methods in the literature. We will complete documentation of the pipeline and protocols, and prepare the final report and dissemination materials.

5 Challenges and Mitigation Strategies

5.1 Difficulties with the EVO Model

During implementation, the extraction of embeddings using the EVO model (specifically, `evo-1-8k-base`) presented significant technical challenges. Although this foundation model leverages advanced signal processing—combining rotational attention and convolutional Hyena blocks—and is designed to capture rich biological patterns, its deployment proved problematic.

Hardware Limitations The most persistent barrier was the insufficient video memory (VRAM) available on the GPUs used. With approximately 6.45×10^9 parameters, the EVO model requires around 25.8 GB of VRAM in full precision (FP32), or approximately 13 GB using mixed precision (FP16/BF16). The primary GPU available (NVIDIA RTX 3060) offered only 12.88 GiB of VRAM, which was insufficient even to load the model weights, let alone support the activations, buffers, and PyTorch overhead. This limitation resulted in repeated `CUDA out of memory` errors, preventing execution.

Software and Architecture Incompatibilities In addition to hardware constraints, errors were encountered due to mismatches between GPU architecture, low-level libraries (such as Triton and FlashAttention), and the EVO model. These included:

- **Triton PTX codegen error:** indicating the JIT compiler failed to generate CUDA code optimized for the available hardware.
- **`TypeError: dot() got an unexpected keyword argument 'trans_b'`:** pointing to an incompatibility between library versions.
- **`invalid argument to exchangeDevice`:** observed when attempting CPU execution, suggesting EVO’s dependencies are tightly coupled to specific NVIDIA GPU features (Ampere or newer).

Together, these errors highlighted that even if memory requirements were met, the lack of perfect alignment between hardware, CUDA version, drivers, and low-level libraries formed an insurmountable barrier to running EVO.

5.2 Strategic Response

Due to the infeasibility of using EVO within the available infrastructure, the following mitigation strategies were adopted:

Validation with Lightweight Methods: To ensure the integrity of the remaining machine learning pipeline, k-mer counting was implemented as an initial feature extraction approach. This low-resource method served as a baseline, confirming that the data loading, SVM training, and evaluation components functioned as intended.

Transition to DNABERT-S: Given the persistent challenges associated with EVO, the project shifted to DNABERT-S as the primary foundation model for DNA sequence analysis. As will be detailed in the following section, DNABERT-S offers several advantages aligned with the project objectives: more manageable computational requirements compared to EVO’s 7 billion parameters, species-aware embedding capabilities particularly relevant for *E. coli*-specific regulatory element optimization, demonstrated effectiveness in few-shot learning scenarios (valuable given the limited experimental data typical in synthetic biology), and robust performance even with dimensionality reduction, enabling efficient processing.

Hybrid Implementation Strategy: The adopted approach leverages DNABERT-S for sophisticated DNA sequence embedding generation while maintaining traditional machine learning methods (SVMs and neural networks) for final prediction tasks. This strategy combines the biological insights captured by the species-aware foundation model with computationally efficient downstream processing.

Although the inability to use EVO was a setback given its multi-modal capabilities, the transition to DNABERT-S represents a strategic adaptation that maintains the sophistication of foundation model approaches while addressing practical implementation constraints. The species-aware nature of DNABERT-S embeddings may prove particularly advantageous for the *E. coli*-focused regulatory element optimization objectives of this work.

DNABERT-S represents an advanced genomic foundation model that extends DNABERT-2’s architecture to generate species-discriminative DNA embeddings. This model specifically addresses the limitations of traditional methods like Tetranucleotide Frequency (TNF) and DNA2Vec in distinguishing genetic content between species, which is particularly crucial for metagenomic analysis and synthetic biology applications [28].

The model incorporates two key innovations to achieve species sensitivity. First, the **Manifold Instance Mixup (MI-Mix)** technique enhances contrastive learning by mixing hidden representations of DNA sequences at random layers during training, forcing the model to discern subtle species-specific patterns. Second, the **Curriculum Contrastive Learning (C²LR)** approach implements a two-phase training strategy. The initial phase employs weighted SimCLR with emphasis on "hard negatives" to improve discrimination, while the second phase integrates MI-Mix to further refine species-level distinctions [28].

For training, DNABERT-S was fine-tuned on approximately 2 million non-overlapping 10 kilobase sequence pairs from diverse microbial species including viruses, fungi, and bacteria. This carefully curated dataset ensures the model learns meaningful biological patterns rather than sequence artifacts, with particular relevance to microbial synthetic biology applications [28].

In benchmark evaluations, DNABERT-S demonstrates exceptional performance, doubling species clustering accuracy (achieving 54% Adjusted Rand Index compared to 26% for baseline methods) on the CAMI II standard. The model also shows remarkable few-shot learning capabilities, achieving superior classification with only 2 samples per species, and maintains robust performance even when embeddings are compressed to 32 dimensions. These characteristics make DNABERT-S particularly valuable for metagenomic binning and other applications where labeled data may be scarce [28].

For synthetic biology applications like those in your project, DNABERT-S offers several advantages. Its species-aware embeddings enable more precise design of *E. coli*-optimized regulatory elements (promoters and RBS sequences), while its few-shot learning capabilities address the common challenge of limited experimental data in synthetic biology workflows. The model’s ability to maintain performance with dimensionality-reduced embeddings also makes it computationally practical for integration into design pipelines [28]).

6 Data PreProcessing

From the original dataset by Kosuri et al., a subset of variables was selected for preprocessing. The most relevant were: `prot` (the target variable, representing sfGFP protein expression), `count.DNA` and `count.RNA` (used for quality control), and a set of boolean quality flags (e.g., `bad.prot`, `min.prot`, `bad.RNA`) indicating unreliable measurements.

The filtering process was based primarily on these flags, which reflect the experimental thresholds and validation criteria described in the supplementary materials of the original study. Constructs with unreliable protein or RNA expression, low DNA/RNA counts, or technical issues with the promoter were excluded. Additionally, constructs lacking protein expression values were removed, as they could not be used in supervised learning.

This filtering ensured that only high-quality, biologically relevant constructs were retained for training, reducing noise and potential bias in downstream modeling steps. Two different standardization methods, `StandardScaler` and `RobustScaler`, were applied to the target variable to account for its non-normal distribution and assess the sensitivity of the models to different scaling approaches.

7 Results

7.1 Introduction and Experimental Setup

The protein expression prediction task was analyzed using a curated dataset of 9,336 high-quality samples, each represented by 768-dimensional DNABERT embeddings. The target variable exhibited substantial variability, with the following statistical characteristics:

- Minimum value: 1357.15 units

- Maximum value: 204059.95 units
- Mean: $\mu = 61077.16$ units
- Standard deviation: $\sigma = 64770.39$ units
- Range: 202702.80 units

To address this prediction challenge, two distinct machine learning architectures were implemented and systematically compared:

1. **Support Vector Machines (SVM)** for regression, implemented using Scikit-learn.
2. **Feedforward Neural Networks (FNN)**, developed in PyTorch.

Each model architecture was initially evaluated with two alternative preprocessing strategies (StandardScaler and RobustScaler), resulting in four experimental configurations for comprehensive performance assessment. This design enabled rigorous comparison of both model architectures and preprocessing approaches on the protein expression prediction task.

For the initial exploration of hyperparameters for the SVM models, a **RandomizedSearchCV** approach was employed to mitigate the extensive computational time often associated with exhaustive Grid Search. This method was preferred for its efficiency in sampling a large parameter space. A search space for the C parameter was defined, typically ranging from 0.001 to 10000, and for gamma from 0.00001 to 10, consistently using an RBF kernel. The initial FNN models, developed in PyTorch, began with a simpler architecture, featuring a single hidden layer of 128 neurons, a fixed learning rate of 0.001, and were trained for 100 epochs with a batch size of 64. Both SVM and FNN models were initially evaluated using 5 folds of cross-validation. Subsequently, for more robust evaluation, the cross-validation folds for SVMs were increased to 10.

7.2 Initial Model Performance

Model evaluation was performed through cross-validation and subsequent testing on the complete dataset. The initial performance metrics results on the original protein expression scale are presented in Table 1. It is important to note that the values for SVMs presented in this table reflect optimized parameters chosen after extensive testing, including addressing initial overfitting issues (detailed in Section 2.3).

The visual comparison of predicted versus actual protein expression values for these initial models is presented in Figure 1.

The initial FNN model with RobustScaler emerged as the best-performing configuration, achieving a coefficient of determination (R^2) of 0.9652 and a Pearson correlation of 0.9829. This result indicated that the model could explain approximately 96.5% of the variance in protein expression data, significantly outperforming the SVM models even in their best configurations.

Performance Comparison

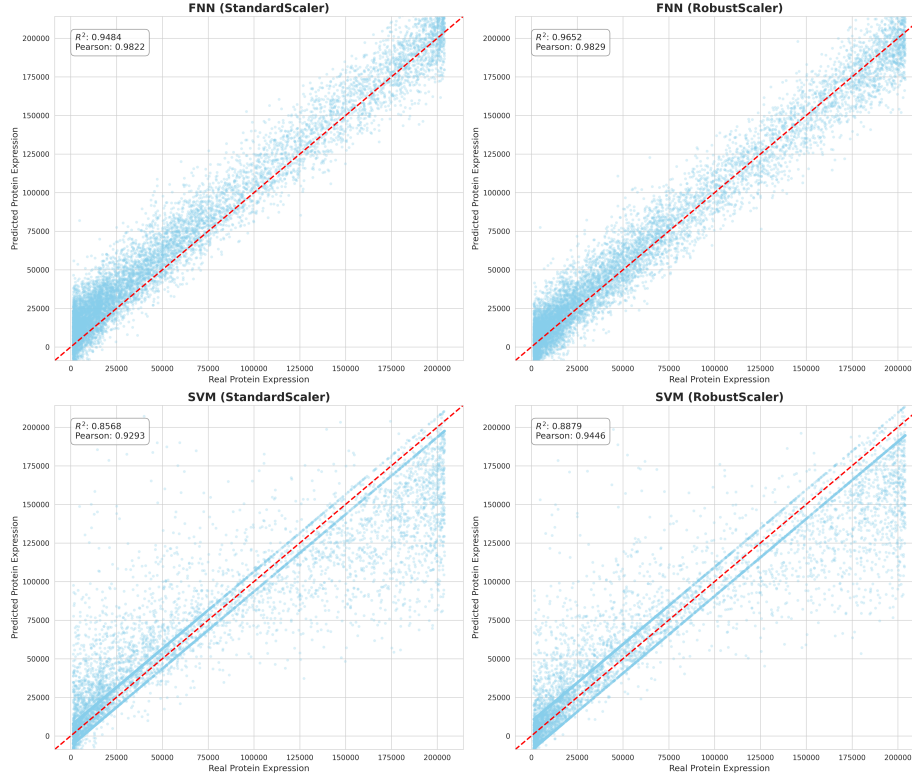


Fig. 1. Scatter plots comparing actual versus predicted protein expression values for the initial FNN and SVM models with StandardScaler and RobustScaler. The red diagonal line represents perfect prediction ($y=x$). R^2 and Pearson correlation coefficients are displayed for each subplot.

7.3 SVM Hyperparameter Optimization Challenges and Revised Performance

The process of optimizing SVM hyperparameters proved to be challenging and time-consuming. Initial attempts with RandomizedSearchCV showed limited improvement over baseline performance. To accelerate the search for optimal hyperparameters, HalvingRandomSearchCV was explored, a technique designed to reduce computational time by iteratively pruning poorly performing candidates across 10 cross-validation folds. However, using an initial `min_resources` parameter set to 4000 (representing the minimum number of samples used for training in the first iteration) within HalvingRandomSearchCV led to severe overfitting. This was evidenced by a stark contrast between high R^2 values (near 0.99) observed when evaluating the model on the full training dataset, versus near-zero

Table 1. Comparison of initial regression model performance

Model/Scaler	R^2	Global Pearson	Global MAE	Global MAE %	RMSE	Global RMSE %
SVM (StandardScaler)	0.8568	0.9293	15066.74	24.67	24512.54	40.13
SVM (RobustScaler)	0.8879	0.9446	14285.41	23.39	21681.21	35.50
FNN (StandardScaler)	0.9484	0.9822	11597.33	18.99	14715.46	24.09
FNN (RobustScaler)	0.9652	0.9829	9313.86	15.25	12083.04	19.78

or negative R^2 values on unseen data during cross-validation. This discrepancy indicated that the model was memorizing the training data rather than learning generalizable patterns.

Reducing `min_resources` to 500 resolved this overfitting issue, allowing `HalvingRandomSearchCV` to identify more robust hyperparameters that improved the model’s generalization capabilities. The updated performance metrics for the SVM models with these more appropriate parameters are reflected in Table 1, representing the best performance achieved for SVMs in this study. While this adjustment corrected the overfitting, the overall predictive capacity of the SVMs, despite extensive testing of various parameter configurations, remained inferior compared to the initial FNNs. This could be attributed to inherent limitations of SVMs in capturing highly complex, high-dimensional relationships, or potentially to the vast parameter space requiring an even more exhaustive search. However, given the superior performance of the FNN models even in their initial phases, subsequent optimization efforts were strategically focused on the latter.

7.4 Enhanced FNN Performance

Given the superior initial performance of the FNN model, particularly when combined with RobustScaler (as shown in Table 1), further optimization efforts were concentrated on this architecture. The objective was to enhance its predictive accuracy and, crucially, to increase the robustness and generalization capabilities through advanced training strategies. The following key modifications were implemented, targeting the FNN (RobustScaler) model:

- **Deeper Architecture:** The FNN was expanded from a single hidden layer to a two-hidden-layer configuration. The initial FNN had a single hidden layer of 128 neurons. This enhanced structure now features two hidden layers with dimensions of 256 and 128 neurons respectively. This deeper structure aimed to enable the model to learn more hierarchical and complex representations from the 768-dimensional DNABERT embeddings.
- **Regularization Techniques:** To combat overfitting, which is a common challenge in complex neural networks, two regularization methods were applied:
 - **Dropout:** A dropout rate of 0.2 was applied after each hidden layer, randomly deactivating neurons during training. This prevents over-reliance on specific neurons and encourages the network to learn more robust features.

- **L2 Regularization (Weight Decay):** A weight decay penalty of 1×10^{-5} was added to the Adam optimizer. This technique penalizes large weight values, discouraging the model from assigning excessive importance to individual features and promoting smoother weight distributions.
- **Learning Rate Scheduling:** A ReduceLROnPlateau scheduler was implemented to dynamically adjust the learning rate during training. This scheduler monitors the validation loss and reduces the learning rate by a factor of 0.5 if the loss does not improve for a patience of 10 epochs, down to a minimum learning rate of 1×10^{-6} . This adaptive learning rate strategy allows for faster initial convergence and finer tuning in later stages of training.
- **Early Stopping:** The training process incorporated an early stopping mechanism with a patience of 25 epochs. This strategy monitors the validation loss and halts training if no improvement is observed within the specified patience window, preventing overfitting by capturing the model state with optimal generalization performance.

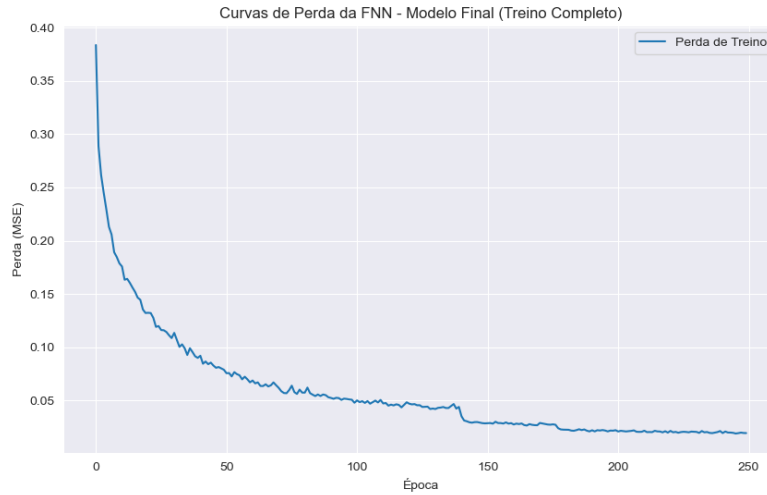


Fig. 2. Training loss curve for the final optimized Feedforward Neural Network (FNN) model, showing Mean Squared Error (MSE) over epochs. The curve illustrates the model's convergence during training.

The performance of the enhanced FNN (RobustScaler) model is summarized in Table 2.

Figure 2 illustrates the training loss curve for the optimized FNN model, demonstrating its convergence during the training process. To conserve space in this report, comprehensive plots detailing both training and validation loss

curves, along with other training progression metrics, have been made available in the project’s GitHub repository (<https://github.com/joaolima88/Projeto-Bioinformatica>).

Table 2. Performance of the Enhanced FNN (RobustScaler) Model

Model/Scaler	R^2	Global Pearson	Global MAE	Global MAE %	RMSE	Global RMSE %
FNN (RobustScaler) Enhanced	0.9893	0.9950	5150.18	8.43	6705.52	10.98

8 Discussion

8.1 Impact of Preprocessing Methods

The results clearly demonstrate the importance of appropriate data preprocessing method selection. **RobustScaler consistently outperformed StandardScaler** in both model types. For SVM models, the use of RobustScaler, even after extensive hyperparameter tuning, yielded better results than StandardScaler (increasing R^2 from 0.8568 to 0.8879). This improvement can be attributed to RobustScaler’s robustness against outliers, using the median and interquartile range instead of mean and standard deviation. Given that protein expression data often exhibit considerable variability and potential outliers, this more robust approach proved particularly suitable.

Even for FNN models, which already showed high performance with StandardScaler, RobustScaler provided consistent initial improvements, increasing R^2 from 0.9484 to 0.9652 and reducing the percentage MAE from 18.99% to 15.25%. This consistent benefit of RobustScaler across both model architectures underscores its effectiveness for this dataset.

8.2 Superiority of Feedforward Neural Networks and Impact of Optimization

The comparison between the two model types reveals a clear superiority of FNNs over SVMs for this specific task. This difference can be explained by several reasons:

Non-linear Modeling Capacity: FNNs are inherently capable of capturing complex non-linear relationships between DNABERT embedding features and protein expression. Considering that regulatory mechanisms governing gene expression are intrinsically non-linear and involve multiple interactions between promoter elements and RBS, this capacity is fundamental.

High-Dimensional Data Processing: With 768 features per sample, DNABERT embeddings constitute a high-dimensional space. FNNs are particularly effective at finding useful patterns in these complex spaces, especially when combined with appropriate regularization techniques.

Architectural Flexibility: The FNN architecture allows the incorporation of multiple hidden layers and non-linear activation functions, enabling the learning of hierarchical data representations that can capture both local and global patterns in DNA sequences.

The initial FNN models already demonstrated strong performance. Indeed, even the initial FNN configurations significantly outperformed the best SVM results obtained after extensive hyperparameter optimization (as shown in Table 1). This striking difference led to the strategic decision to concentrate further optimization efforts on the FNN (RobustScaler) model, which had shown the most promising initial capabilities.

Through this focused optimization, the FNN’s predictive capabilities were substantially elevated. The expansion to a deeper architecture (two hidden layers) allowed the model to learn more intricate and abstract representations of the input features. The combined application of Dropout and L2 Regularization effectively mitigated overfitting, ensuring that the model generalized well to unseen data. Crucially, the implementation of a Learning Rate Scheduler and Early Stopping facilitated a more refined training process, allowing the model to converge to a more optimal state and preventing training beyond the point of best generalization.

These optimization strategies resulted in a dramatic improvement in the FNN’s performance metrics. The R^2 Global increased from an already strong 0.9652 to an outstanding 0.9893, indicating that nearly 99% of the variance in protein expression can now be explained by the model. The Pearson correlation improved to 0.9950, reflecting an almost perfect linear relationship between predicted and actual values. Furthermore, the MAE Global was reduced significantly from 9313.86 to 5150.18, corresponding to a mean absolute error of just 8.43% of the average protein expression. The RMSE Global also saw a proportional reduction to 6705.52 (10.98% of the mean value).

This level of performance solidifies the FNN as the superior model for this prediction task. While SVMs were extensively explored, testing multiple configurations and addressing initial overfitting challenges, their inherent limitations in capturing the complex, high-dimensional relationships within DNABERT embeddings, coupled with challenges in hyperparameter optimization, rendered them less effective compared to deep learning architectures like FNNs. It is acknowledged that even more exhaustive exploration of SVM’s vast parameter space might yield marginal further improvements, but the FNN’s consistently strong and highly optimizable performance provided a clear and efficient path forward for achieving the highest accuracy.

8.3 Implications for Genetic Construct Engineering

The highly accurate results obtained with the optimized FNN model have significant implications for rational genetic construct engineering. With a mean error of only 8.43% (reduced from 15.25%), the developed model enables two key applications in synthetic biology. First, it facilitates virtual screening by allowing in silico evaluation of thousands of promoter-RBS combinations prior to

experimental validation, which dramatically reduces both the time and cost associated with traditional trial-and-error approaches. Second, the model provides a powerful tool for directed optimization, as it can systematically guide the design of new genetic constructs by identifying combinations with high expression potential based on their DNA sequence features. This predictive capability represents a paradigm shift from empirical testing to computationally-driven design in genetic engineering workflows.

References

1. Akdemir, B., Polat, K., Güneş, S.: Prediction of E.Coli promoter gene sequences using a hybrid combination based on feature selection, fuzzy weighted pre-processing, and decision tree classifier. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) Knowledge-Based Intelligent Information and Engineering Systems. Lecture Notes in Computer Science, vol. 4692, pp. 125–131. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
2. Carrier, T.A., Keasling, J.D.: Library of synthetic 5 secondary structures to manipulate mrna stability in *Escherichia coli*. *Biotechnology Progress* **15**(1), 58–64 (1999). <https://doi.org/10.1021/bp9801143>, <https://doi.org/10.1021/bp9801143>
3. Chen, Y., Banerjee, D., Mukhopadhyay, A., Petzold, C.J.: Systems and synthetic biology tools for advanced bioproduction hosts. *Current Opinion in Biotechnology* **64**, 101–109 (2020). <https://doi.org/10.1016/j.copbio.2019.12.007>, <https://www.sciencedirect.com/science/article/pii/S0958166919301454>
4. Chubiz, L.M., Rao, C.V.: Computational design of orthogonal ribosomes. *Nucleic Acids Research* **36**(12), 4038–4046 (2008). <https://doi.org/10.1093/nar/gkn354>, <https://doi.org/10.1093/nar/gkn354>
5. Ellis, D.I., Goodacre, R.: Metabolomics-assisted synthetic biology. *Current Opinion in Biotechnology* **23**(1), 22–28 (2012). <https://doi.org/10.1016/j.copbio.2011.10.014>, <https://www.sciencedirect.com/science/article/pii/S0958166911007105>
6. Greener, J.G., Kandathil, S.M., Moffat, L., Jones, D.T.: A guide to machine learning for biologists. *Nature Reviews Molecular Cell Biology* **23**(1), 40–55 (2022). <https://doi.org/10.1038/s41580-021-00407-0>, <https://doi.org/10.1038/s41580-021-00407-0>
7. Isaacs, F.J., et al.: Engineered riboregulators enable post-transcriptional control of gene expression. *Nature Biotechnology* **22**, 841–847 (2004)
8. Johns, N.I., Gomes, A.L.C., Yim, S.S., Yang, A., Blazejewski, T., Smillie, C.S., Smith, M.B., Alm, E.J., Kosuri, S., Wang, H.H.: Metagenomic mining of regulatory elements enables programmable species-selective gene expression. *Nature Methods* **15**, 323–329 (2018)
9. de Jongh, R.P.H., van Dijk, A.D.J., Julsing, M.K., Schaap, P.J., de Ridder, D.: Designing eukaryotic gene expression regulation using machine learning. *Trends in Biotechnology* **38**(2), 191–201 (2020)
10. Jumper, J., et al.: Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021). <https://doi.org/10.1038/s41586-021-03819-2>
11. Kolides, A., Nawaz, A., Rathor, A., Beeman, D., Hashmi, M., Fatima, S., et al.: Artificial intelligence foundation and pre-trained models: Fundamentals, applications, opportunities, and social impacts. *Simulation Modelling Practice and Theory* **126**, 102754 (2023). <https://doi.org/10.1016/j.simpat.2023.102754>, <https://www.sciencedirect.com/science/article/pii/S1569190X2300031X>
12. Kosuri, S., Goodman, D.B., Cambray, G., Mutalik, V.K., Gao, Y., Arkin, A.P., Endy, D., Church, G.M.: Composability of regulatory sequences controlling transcription and translation in *Escherichia coli*. *Proceedings of the National Academy of Sciences of the United States of America* **110**(34), 14024–14029 (2013). <https://doi.org/10.1073/pnas.1301301110>, <https://doi.org/10.1073/pnas.1301301110>

13. Lee, S.Y., Mattanovich, D., Villaverde, A.: Systems metabolic engineering, industrial biotechnology and microbial cell factories. *Microbial Cell Factories* **11**, 156 (2012). <https://doi.org/10.1186/1475-2859-11-156>, <https://doi.org/10.1186/1475-2859-11-156>
14. Lv, X., Hueso-Gil, A., Bi, X., Wu, Y., Liu, Y., Liu, L., Ledesma-Amaro, R.: New synthetic biology tools for metabolic control. *Current Opinion in Biotechnology* **76**, 102724 (2022). <https://doi.org/10.1016/j.copbio.2022.102724>, <https://www.sciencedirect.com/science/article/pii/S0958166922000581>
15. Meng, H., Ma, Y., Mai, G., Wang, Y., Liu, C.: Construction of precise support vector machine based models for predicting promoter strength. *Quantitative Biology* **5**, 90–98 (2017)
16. Nguyen, E., Poli, M., Durrant, M.G., Kang, B., Katrekar, D., Li, D.B., et al.: Sequence modeling and design from molecular to genome scale with Evo. *Science* **386**(6723), eado9336 (2024). <https://doi.org/10.1126/science.ado9336>
17. Oesterle, S., Wuethrich, I., Panke, S.: Toward genome-based metabolic engineering in bacteria. In: Sariaslani, S., Gadd, G.M. (eds.) *Advances in Applied Microbiology*, vol. 101, pp. 49–82. Academic Press (2017). <https://doi.org/10.1016/bs.aambs.2017.07.001>, <https://www.sciencedirect.com/science/article/pii/S0065216417300394>
18. Ortuño-Fajardo, M.P., Chacón-Halabi, J.R., Flores-Espinoza, M.P., Aguilar-Bravo, R.: Biología sintética en la ingeniería de rutas metabólicas de microorganismos para la obtención de compuestos de interés para la industria alimentaria. *Revista Tecnología en Marcha* **34**(1), 69–79 (2021). <https://doi.org/10.18845/tm.v34i1.4830>, https://revistas.tec.ac.cr/index.php/tec_marcha/article/view/4830
19. Pflieger, B.F., Pitera, D.J., Smolke, C.D., Keasling, J.D.: Combinatorial engineering of intergenic regions in operons tunes expression of multiple genes. *Nature Biotechnology* **24**, 1027–1032 (2006)
20. Poli, M., Massaroli, S., Nguyen, E., Fu, D.Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., Re, C.: Hyena hierarchy: Towards larger convolutional language models. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) *Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 202, pp. 28043–28078. PMLR (2023). <https://doi.org/10.48550/arXiv.2302.10866>, <https://proceedings.mlr.press/v202/poli23a.html>
21. Porcello, J.C.: Designing and implementing svms for high-dimensional knowledge discovery using fpgas. In: 2019 IEEE Aerospace Conference. pp. 1–8 (2019). <https://doi.org/10.1109/AERO.2019.8741916>
22. Pérez González, A.: Synthetic Biology Tools for the Study of Relevant Factors in the Control of Transgene Expression. Ph.D. thesis, Universidad Politécnica de Madrid (2018). <https://doi.org/10.20868/UPM.thesis.52923>, <https://oa.upm.es/52923/>
23. Radivojević, T., Costello, Z., Workman, K., Garcia Martin, H.: A machine learning Automated Recommendation Tool for synthetic biology. *Nature Communications* **11**(1), 4879 (2020). <https://doi.org/10.1038/s41467-020-18008-4>, <https://doi.org/10.1038/s41467-020-18008-4>
24. Romero, E., Toppo, D.: Comparing support vector machines and feed-forward neural networks with similar parameters. In: Corchado, E., Yin, H., Botti, V., Fyfe, C. (eds.) *Intelligent Data Engineering and Automated Learning – IDEAL 2006*. pp. 90–98. Springer Berlin Heidelberg, Berlin, Heidelberg (2006). https://doi.org/10.1007/11875581_11

25. Xu, P., Li, L., Zhang, F., Stephanopoulos, G., Koffas, M.: Improving fatty acids production by engineering dynamic pathway regulation and metabolic control. *Proceedings of the National Academy of Sciences of the United States of America* **111**(31), 11299–11304 (2014). <https://doi.org/10.1073/pnas.1406401111>, <https://doi.org/10.1073/pnas.1406401111>
26. Zhang, M., Holowko, M.B., Hayman Zumpe, H., Ong, C.S.: Machine learning guided batched design of a bacterial ribosome binding site. *ACS Synthetic Biology* **11**(7), 2314–2326 (2022)
27. Zhang, X., Lin, Y., Wu, Q., Wang, Y., Chen, G.Q.: Synthetic biology and genome-editing tools for improving pha metabolic engineering. *Trends in Biotechnology* **38**(7), 689–700 (2020). <https://doi.org/10.1016/j.tibtech.2019.10.006>, <https://www.sciencedirect.com/science/article/pii/S0167779919302446>
28. Zhou, H., Ji, Y., Chen, J., Wang, Z., Kong, H., Ding, J., Dai, H., Yang, Y.: Dnabert-s: Adapting dnabert for species-aware dna language modeling. *arXiv preprint arXiv:2402.08777* (October 2024), available at: <https://arxiv.org/abs/2402.08777>