ASHESI UNIVERSITY

USE OF MACHINE LEARNING FOR THE OPTIMIZATION OF GENETIC CIRCUITS IN SYNTHETIC BIOLOGY: FOCUSING ON PROMOTER PREDICTION FOR GENE EXPRESSION IN ESCHERICHIA COLI

UNDERGRADUATE THESIS

B.Sc. Computer Science

Nice Cailie Ineza

2024

ASHESI UNIVERSITY

Use Of Machine Learning for The Optimization of Genetic Circuits in Synthetic Biology: Focusing On Promoter Prediction For Gene Expression In Escherichia Coli

UNDERGRADUATE THESIS

Undergraduate Thesis submitted to the Department of Computer Science and Information System, AshesiUniversity College in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science.

Nice Cailie Ineza

August 2024

DECLARATION

I hereby declare that this undergraduate thesis is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……

Candidate's Name:

……

Date: ……

I hereby declare that preparation and presentation of this undergraduate thesis were supervised in accordance with the guidelines on supervision of undergraduate thesis laid down by Ashesi University College.

Supervisor's Signature:

……

Supervisor's Name:

……

Date: ……

Acknowledgement

I am grateful to the Almighty God for the divine spirit of hardworking and courage put in me.

My deepest gratitude goes to my primary supervisor Dr. Elena Rosca, for her patience and belief that I could do a good project, and for encouraging me into taking this project despite my limited knowledge of the field. As well, as for acknowledging my limitations, and give me guidance, critics, feedback during the whole project.

Acknowledging the divine motivation and courage instilled within me, I express profound gratitude to my primary supervisor, Dr. Elena Rosca, for her unwavering support, guidance, and belief in my project's potential. Her constructive feedback and encouragement were pivotal in navigating the challenges of the project.

The technical and logistical support from the University's Department of Computer Science, particularly through our update presentations, were instrumental. I am thankful to Dr. David Ebo and Dr. Govindha Yeluripati for their consistent engagement and insightful feedback on my presentations. Dr. Tatenda Kavu's timely advice and follow-up were crucial in resolving project challenges.

Acknowledgment is also due to Dr. Eric Ocran for his thorough review and Prof. Obiri for his regular check-ins. My friends deserve credit for their camaraderie and for pushing me to maintain momentum, even amidst stress.

Lastly, my family, especially my parent's constant support, prayers, and confidence-building words were invaluable. This acknowledgment is a tribute to the collective efforts and encouragements that have been vital to the completion of my undergraduate thesis project.

Abstract

This study explores the application of Machine Learning (ML) in optimizing genetic circuit design in Synthetic Biology, particularly focusing on predicting specific promoter for gene expression in Escherichia Coli. Despite the potential of ML, current methods rely heavily on trial-and-error, which is inefficient and costly. The research employs various ML models, including Genetic Algorithms, Support Vector Machines, and Neural Networks, alongside computational algorithms like Boyer-Moore-Horspool's algorithm, to predict promoter efficacy and identify optimal promoter-gene configurations. Utilizing datasets from databases such as RegulonDB, EcoCyc, and PRODORIC, the study validates its findings through a combination of literature cross-checks and model performance metrics. The resulting model achieved a high accuracy in predicting promoter efficacy, with a 92% success rate in identifying optimal configurations. The findings suggest that incorporating transcription unit data significantly improves prediction accuracy, demonstrating the potential of ML in advancing synthetic biology towards more precise and efficient genetic circuit design.

Table of Contents

List of Figures

Chapter 1: Introduction

Synthetic biology stands at the forefront of scientific innovation, aiming to redefine the boundaries of medicine, energy production, and environmental conservation through the design and engineering of biological systems. Tracing its roots back to 1970 by the pioneering geneticist Waclaw Szybalski, the discipline has evolved, underpinned by advancements in DNA sequencing and synthesis technologies [1]. At its core, synthetic biology seeks to harness the inherent capabilities of cells to navigate, communicate, and organize into complex structures by manipulating genetic circuits. These circuits, essential for various biological functions, represent a nexus of potential for synthetic biology to engineer life with precision and purpose [3][2]. Yet, the inherent complexity of biological systems presents a challenge, notably in the design and optimization of genetic circuits, where the predictive mapping of components to functions remains elusive for different organisms.

The most used organisms in SB are the bacteria *Escherichia coli*, *Bacillus subtilis* and the *yeast Saccharomyces cerevisiae*. The main reason for their popularity is that these are model organisms widely studied in the laboratory and for which an ample catalogue of molecular tools is available. It is possible to characterize and predict their physiology very accurately, benefitting also the understanding of the performance of SB circuits and pathways [4]. The advent of Machine Learning (ML) offers a promising lens through which to view and tackle these challenges. By applying ML techniques, this project aims to streamline the genetic circuit design process, particularly focusing on the optimization of promoter sequences to control gene expression accurately and efficiently. This approach is poised to mitigate the reliance on traditional, labor-intensive methodologies, paving the way for more rapid and cost-effective development cycles in synthetic biology [3].

1.1. Problem Statement

Despite the potential of Machine Learning to optimize genetic circuit design, there exists a gap in the application of these techniques in synthetic biology. The current and available methods rely heavily on trial-and-error and extensive experimental validation, which is time consuming and expensive, limiting efficiency and scalability. While significant progress has been made in genetic circuit design, there remains a gap between desired circuit functionalities and the strategies employed to achieve them. Genetic circuit design remains one of the most challenging aspects of genetic engineering [3]. The inherent complexity of biological systems poses challenges in elucidating the intricate relationships between genetic components and desired phenotypic behaviours. This study aims to investigate how Machine Learning can be applied to the optimization of genetic circuits in synthetic biology, with the goal of designing circuits that are not only functionally effective but also biologically interpretable and applicable especially for the *Escherichia Coli* organism.

1.1.1. Study Background

In Synthetic biology, genetic circuits enable the manipulation and control of the biological process within living cells. These circuits that can be explained using an example of a typical electric circuit are composed of genetic elements such as promoters, which serves a pivotal role in gene expression regulation, analogous to switches' function in electrical circuits.

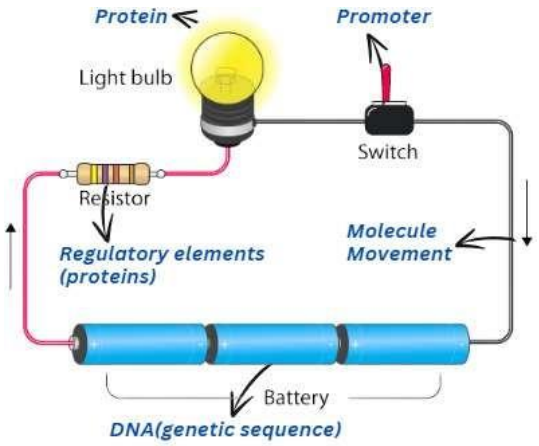

Figure 1. Simple representation of genetic circuit comparing to an electric circuit

Promoters are sequences that initiate the transcription by *Ribonucleic Acid* (RNA) *polymerase*, thereby controlling when and how long a gene is expressed. In prokaryotes like *Escherichia Coli (E. Coli),* promoters are important for their role in governing metabolic pathways and cellular responses to environment stimuli.

1.2. Objectives and Research Questions

This study leverages ML to advance the design and optimization of genetic circuits in synthetic biology, with a specific focus on promoter sequence efficacy. The following objectives frame the research:

1. To develop a machine learning model capable of predicting the most effective promoter sequences for desired gene expression outcomes. This raises the question:
    - How can ML algorithms be tailored to accurately predict promoter efficacy in driving specific gene expression patterns?
2. To design an optimization algorithm that identifies the best promoter-gene configurations for achieving specified synthetic biology applications. This leads to the inquiry:
    - What computational strategies can be employed to sift through the myriad promoter gene configurations to identify those most conducive to desired genetic circuit outcomes?

1.3. The Significance of the Study

The integration of ML into Synthetic Biology represents a significant leap forward in the discipline, fundamentally altering the landscape of genetic circuit design and optimization for organisms like *E. coli*. By facilitating the precise prediction and selection of promoter sequences, this research has the potential to significantly streamline the development of biological systems, reducing the timeline and resource expenditure associated with developing biological systems.

Moreover, the project aims to enhance our understanding of gene regulation mechanisms, contributing valuable insights to the field of synthetic biology for both new learners and experienced learners.

Following this introduction, the literature review will explore current methodologies and challenges in genetic circuit design, emphasizing the role of ML in overcoming these hurdles. It will set the groundwork for understanding the transformative impact of the proposed ML-based optimization framework on synthetic biology's capability to engineer life with unparalleled precision and efficiency.

Chapter 2: Literature Review

2.1. The Story of genetic circuit

Synthetic biology aims to design and build novel biological systems with desired functionalities [5]; it has the innovative realm of genetic circuits, complex networks that underpin the functioning of living cells. Genetic circuits, consisting of interacting genes and regulatory elements, control information flow within biological systems, drawing parallels with computer-aided design (CAD) for electronic circuit engineering [6]. Drawing inspiration from computer science and electronics, synthetic gene circuits have been engineered to control information flow within biological systems [7]. For instance, Lucks et al. emphasize that the efficiency of genetic engineering design cycles greatly benefits from understanding biological design principles and utilizing technologies that enhance these cycles. They advocate for the development of biological parts with defined properties like independence, reliability, and tunability, which are essential for creating new pathways with predictable behaviors in cells, highlighting the importance of genetic circuits in synthetic biology [8].

Promoters play a pivotal role in this landscape, acting as the “switches” that initiate the process of gene expression, a fundamental step towards protein synthesis [7]. They govern the rate of gene transcription and, consequently, protein synthesis, acting as the keystones of genetic circuitry [6]. By understanding these circuits, scientists aim to harness biological processes for better applications, from biotherapeutics to environmental monitoring. The capacity to perform computations within living cells promises to revolutionize biotechnology by enhancing existing products and enabling new applications. Short-term benefits include the improvement of bio-based chemical production through timed gene expression during fermentation stages or condition-specific enzyme activation. As circuits evolve, entire algorithms from control theory could be applied to boost biochemical production [3].

2.2. The Importance of promoters

The search for the efficient protein synthesis in synthetic biology hinges on the optimization of promoters. These DNA sequences are very important for kick-starting transcription, the first step in turning genes into proteins. Recent research highlights the importance of understanding and engineering promoters to fine-tune genetic circuits and metabolic pathways, particularly in microorganisms like *Saccharomyces cerevisiae*. Tang et al. discuss the limitations of native promoters in *S. cerevisiae* and the necessity for promoter engineering to create synthetic promoters with improved properties for metabolic engineering applications. They emphasize the role of machine learning in designing better promoters, showcasing the intersection of computational and biological sciences in synthetic biology [9].

Furthermore, Kotopka and Smolke provide an exemplary study on the model-driven generation of artificial yeast promoters, highlighting the power of combining experimental approaches with convolutional neural network models to predict protein expression from promoter sequences. This study not only contributes large sets of novel promoters but also underscores the value of model-guided design in generating useful DNA parts for synthetic biology [10].

These papers collectively underscore the critical importance of promoters in the design and optimization of genetic circuits. However, the difficulty arises from pinpointing promoters capable of optimal performance across different conditions, a task that has been found to be challenging by the intricate nature of biological systems. Traditional methods, often cumbersome and time-consuming, highlight the urgency for novel approaches capable of accelerating promoter optimization [3]. Those challenges in promoter optimization underscore the necessity for innovative approaches such as machine learning.

2.3. Machine learning in genetic circuits design

In the rapidly evolving field of synthetic Biology, ML emerges as a revolutionary tool for genetic circuit design, able to offer a powerful means to navigate the complexities inherent in this

domain. The intersection of ML and synthetic biology offers promising avenues for the design and optimization of genetic circuits. ML algorithms can analyze vast datasets, learning from the complexities of genetic sequences to predict the most effective promoters for specific functions. This approach could drastically reduce the reliance on experimental trial and error, streamlining the development of genetic circuits tailored for specific functions [11]. This approach not only enhances precision in selecting promoters but also significantly reduces the experimental burden traditionally associated with promoter optimization [12].

Recent studies have showcased the application of ML in genetic circuit design, highlighting varied methodologies ranging from genetic algorithms to deep learning. The exploration of ML in synthetic biology has unearthed methodologies that revolutionize how we approach genetic circuit optimization. There are studies that have employed various ML models, including random forests and neural networks, to predict gene expression levels with impressive accuracy, and other studies that have successfully shown the application of ML in identifying optimal promoters.

Building on the foundational work of Aromolaran et al., who explored the application of machine learning (ML) in predicting gene essentiality, we see a clear path towards harnessing ML's power to optimize genetic circuits. While focusing on essential genes, this study underlines the importance of discovering relevant features for classification and the generalizability of prediction models, which are paramount in the context of promoter optimization in genetic circuits. This emphasis on essential genes sets the stage for a deeper dive into how ML can enhance promoter optimization in genetic circuits, especially in predicting conditionally essential genes_a frontier where ML's potential to decipher promoter efficiency under specific conditions becomes evident [13].

Following suit, Dixit & Prajapati expanded on the adaptive process of machine learning in bioinformatics, specifically addressing the classification of gene sequences. Their study not only reinforces the utility of ML in distinguish between healthy and diseased genes but also illuminates

it applicability in promoter optimization within genetic circuits. By leveraging ML techniques, we can refine our ability to select promoters that align with desired gene expression profiles, thereby paving the way for more precise genetic circuit design and function. This seamless transition from gene classification to promoter optimization showcases the interconnectedness of these processes and the pivotal role ML play in bridging the gap between theoretical knowledge and practical applications in Synthetic Biology [14].

Furthering our understanding, Vahid et al. [15] employed an Expectation Maximization and Support Vector Machine classifier (EMSVM) for promoter detection in DNA sequences. Their method comprises two phases: initially, data clustering is achieved via expectation maximization (EM), followed by classification with support vector machines (SVM). This approach demonstrated exceptional accuracy in identifying promoters. Yang et al. [7] provided a comprehensive review of the application of the machine learning algorithms in DNA sequence data mining, with a particular focus on DNA sequencing, classification, clustering, and pattern mining. This review not only elaborates on the various machine learning algorithms used for DNA data analysis but also suggests the potential of these methods for identifying and predicting enhancer-promoter interactions. Such interactions are critical for the transcriptional regulation of gene expression, thus highlighting the relevance of ML in optimizing promoter selection to achieve efficient gene expression in synthetic biology [7].

Saltepe et al. explore the combination of machine learning (ML) and genetic circuits to produce rapid-response sensors. In their work, they enhance the features of whole cell biosensors (WCBs) using neural network-based architecture to improve rapidness, simplicity, and accuracy in biosensing. They demonstrate which ML systems are best suited for providing immediate responses when dealing with genetic circuits, emphasizing the need to tune circuits to meet specific application requirements [12].

An additional paper explains what implications to consider when developing a machine-learning-based system for genetic circuit optimization. Zhu et al. [11] shared that one needs to understand the mathematical models and any other ML models that come with the design of genetic circuits. The paper applied some machine learning techniques to construct mathematical models for predicting gene expression. Data preparation techniques are stressed to ensure high accuracy in the ML models. An experiment using green fluorescent protein (GFP) demonstrates that a classifier model based on GFP can predict the intensity of fluorescence emitted by GFP when it is excited by light (GFP values) with 100% accuracy, enabling the identification of various synthetic gene circuits. The paper illustrates the potential of different machine learning techniques such as random forest (RF), support vector machines (SVM), and artificial neural networks (ANN) for constructing models to predict gene expression in genetic circuit designs.[11]

These studies underscore the potential of machine learning in advancing our understanding and capabilities in genetic circuit design. These approaches have demonstrated success in predicting promoter behavior, yet they also reveal limitations, including the need for extensive datasets and the challenge of translating ML predictions into biological contexts [16].

2.4. Identifying the gap

A critical analysis of existing literature underscores a significant gap in the application of ML for promoter optimization in genetic circuit sequence. The diversity of promoters and the unique contexts in which they operate necessitate a more tailored ML approach. Current methodologies, while promising, often fall short in addressing the nuanced requirements of specific bacterial strains, pointing to the need for innovative solutions that can adapt to these unique biological contexts. Moreover, the complexity of genetic circuits, coupled with the non-linearity of biological processes, poses significant challenges for ML models, which may not fully capture the dynamics of gene expression [16][11].

2.5. The Proposed approach

To address these gaps, the proposed ML methodology aims to harness genetic algorithms and advanced ML models to navigate the vast landscape of genetic sequences, identifying optimal promoters. A study by Santoso et al. on virus prediction using DNA sequences further supports the potential of machine learning and deep learning methods for classifying and predicting based on genetic data. This reinforces the concept that similar methods can be applied to promoter optimization in genetic circuits for synthetic biology [17]. This approach will leverage comprehensive datasets, encompassing a wide array of promoter sequences and their associated output characteristics, to train ML models capable of predicting efficient promoters for varied applications in synthetic biology [12].

The fusion of ML and synthetic biology heralds a new era in the design and development of genetic circuits, offering the potential to revolutionize protein synthesis through the optimization of promoters. This literature review underscores the promise and challenges of this interdisciplinary endeavor, paving the way for further research that could extend beyond promoter optimization to encompass the entirety of genetic circuit design, thereby amplifying the impact of synthetic biology on medicine, industry, and environmental sustainability [5][11]. By focusing on promoter efficiency, particularly within the realm of gene sequence, this research promises to enhance our ability to engineer biological systems with unprecedented precision and efficiency.

Chapter 3: Methodology

The research methodology for the capstone project aims to develop a machine-learning-based framework for optimizing genetic circuits in synthetic biology. This framework is envisioned to facilitate the translation of desired inputs into functional outputs, thereby streamlining the design process of genetic circuits. The project's main objectives include:

1. Developing a machine learning model to predict and choose the most effective promoter sequences for desired gene expression outcomes: this evolved into a model that gives all the details one needs in order to choose an appropriate promoter to use for their specified genes; some of those details include the operon in which the gene they choose is part, the transcription unit, allowing them to easily identify their gene in the circuits or where to put it in the gene circuit, the promoter sequence and the regulator, or what kind of promoter they have and the functionality they can achieve.

2. Designing an optimization algorithm that identifies an appropriate promoter-gene configurations for achieving specified synthetic biology applications. This involved trying to check if, given the confidence level in which we have the promoter, whether it is “strong” or "weak," we are able to use such an indication to give the user an output of what to use without necessarily through all the selections or suggestions. We also looked into doing it in case a user has a specific functionality they want to get without necessarily knowing what genes to use.

3.1. Proposed architecture

The objective of our research is to develop a model that predicts the promoters for specific gene circuits by focusing on promoters that could be compatible with a specific gene in *E. coli* bacteria using machine learning.

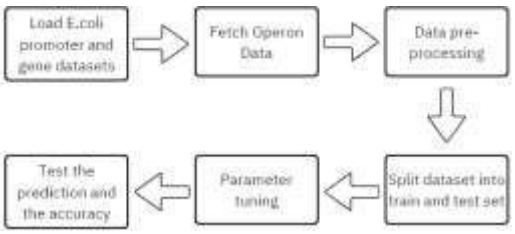

Figure 3 1. Architecture of the system model flow

3.2. Approach

The research leveraged machine learning (ML) techniques to predict the behavior of genetic circuits and optimize their designs. This approach is based on the recognition that ML can learn the complex relationships between genetic components and circuit functionalities, thereby offering a solution to the challenges of predicting behavior in biological systems. Upon our system review and research refinement, we chose three machine learning models and one computational algorithm as our tools for implementation.

3.2.1. Machine Learning

The specific and useful machine learning techniques identified for this project include:

- Genetic Algorithm (GA): This evolutionary algorithm was used to optimize feature selection for classifying gene products based on their functions. They are adaptive heuristic search algorithms based on the principles of natural selection and genetics. They can be mainly used for:
    - Optimizing complex problems by evolving solutions over generations
    - Applying operations like selection, crossover, and mutation to create new populations of solutions
    - Improving genetic circuit designs by selecting the best genes to use for a particular sequence based on a function.

The process involved the application of a Naïve Bayes classifier to the pre-processed gene product and vectorized text data. It helped us improve the selection of features and give a possible

best choice of a gene in order to achieve our second objective. This approach enhanced the predictive performance of the model and also facilitated the recommendation of genes associated with specific functions, thereby aiding in the design of genetic circuits.

1. Support Vector Machine (SVM): It is an algorithm used for classification and regression tasks. For our research, SVM was employed to classify gene functions and regulatory elements based on their features. It helped in finding the optimal hyperplane that separates different classes in the feature space. The training data for the SVM model is training data. The model can be represented by:

$$f(x) = w^T\emptyset(x) + b$$

where w: weight vector, x: feature transformation, b: bias term

The model is trained to find the optimal w and b by minimizing the cost function with regularization, typically:

$$min_{w,b}\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\max(0,1 - y_i(w^T\emptyset(x_i) + b))$$

2. Neural Networks: This is a computational model inspired by the human brain, capable of detecting patterns in complex data. They can be used for:

- Predicting the behavior of genetic circuits by learning from training data.
- Mapping the relationships between genetic components and circuit functionalities.
- Employing layers of interconnected nodes to process and analyze gene expression.

Neural networks learn through a technique called backpropagation, which allows the network to adjust the weights of the hidden layer in order to minimize the value of the error obtained by the cost function. This optimization is done using a method called gradient descent, which calculates the derivative of the loss function with respect to the weights. The cost function used in

building the neural network for the purpose of this work is binary cross-entropy, which is used for binary classification.

3. Clustering and Mapping Technologies: Clustering algorithms, such as K-means and hierarchical clustering, were employed to group genes and classes of operon they belonged to, highlighting other promoters that could be used in their cases. Mapping technologies, such as t-SNE (t-Distributed Stochastic Neighbor Embedding), were used to visualize dimensional gene data, helping to identify patterns and relationships between the components.

3.2.2. Computational Algorithm

Boyer-Moore-Horspool's algorithm: It is an algorithm for finding substrings in strings. This is a string-searching algorithm that we used for:

- Efficiently finding occurrences of patterns within a given text.
- Applying it in the context of our research project helps in identifying some gene sequences within promoter sequences, as some literature suggests, there may be a relationship.
- Enhancing the accuracy of pattern recognition in genetic circuits.

3.2.3 Implementation and Technologies

The development and implementation of the machine learning-based framework were supported by a range of tools and technologies, including:

o Python libraries, like Sklearn, TensorFlow, and PyTorch, for machine learning development. Tensor flow was useful for scalability and automatic differentiation in our computation.

o Docker and Ngrok: to host datasets locally and enable API usage in Google Colab.

3.4. Data Sources

The data used in our research were secondary data from existing databases and literature. Those datasets were:

1. RegulonDB: An online database specializing in the inner workings of gene regulation in Escherichia coli K–12. It provides detailed information on transcriptional regulation, gene organization into operons, and regulatory networks. Just for reference, an **operon** is a functional unit of DNA in prokaryotes, comprising a cluster of genes under the control of a single promoter and regulated together. A **regulator** is a molecule or protein that influences the expression of genes. The database summary:

   - 4748 Genes
   - 2613 Operons
   - 287 Regulons
   - 3742 Transcriptions.

2. EcoCyc: The data focuses on the Escherichia coli K-12 MG1655 strain, offering detailed information on genes, their sequences, functions, and encoded proteins. The information in that database is curated from over 44,000 scientific publications.

3. PRODORIC: Provides comprehensive data on prokaryotic promoters, regulatory interactions, and gene expression.

3.5. Methods for validation

Validation of the developed system will be conducted through a literature cross-check, where predictions are confirmed with existing papers or published work. We:

- Compare predicted circuit behavior with actual experimental results.
- Analyze the biological mechanisms underlying circuit performance.
- Evaluate the generalizability of the model.

Chapter 4: Experiments and results

4.1. Databases hosting

To start, the choice of datasets and the determination of relevant versus irrelevant information were paramount. Hosting the database, RegulonDB, locally preceded the creation of its API. The diagram below shows the database function and set-up:

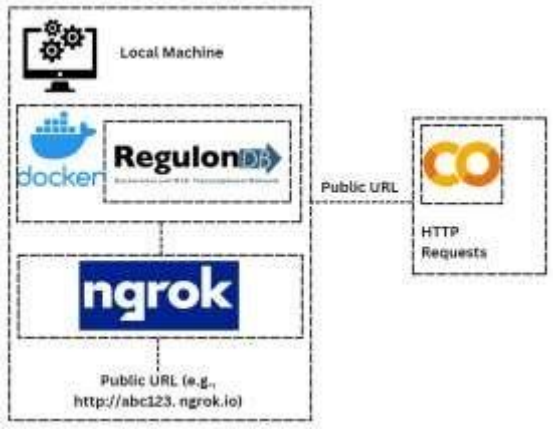

Figure 4. 1. Diagram of datasets hosting and deployment

1. RegulonDB in a Docker container:

    1. Docker is used to host RegulonDB on the local computer. It creates an environment that contains the RegulonDB database.

    2. RegulonDB: the database is running inside the Docker container and accessible via a specific port on the local machine (e.g.: ‘localhost:7000’).

2. Ngrok:

    1. Ngrok is used to create a secure tunnel to the local machine, exposing the local port where RegulonDB is hosted to the internet. It is connected via the command line: *docker run --net=host -it -e NGROK_AUTHTOKEN=**Generated_key** ngrok/ngrok: latest **http 7000***

3. Public URL: ngrok provides a public URL that forwards requests to ‘localhost:7000’, making the database accessible.

4. Google Colab:

    1. Colab is used to run your code and interact with RegulonDB. Colab, then use the public URL provided by ngrok to send a query.

    2. HTTP Requests: Colab sends HTTP requests to the public URL to access the RegulonDB API.

Like any machine learning analysis, the first step is data pre-processing. The pre-processing of raw data to obtain useful information that is feasible for analysis is crucial, as it determines the efficiency of analysis as well as the quality of the output. Next to the pre-processing, we conducted model training, evaluation, and final visualization.

4.2. Dataset preparation

In the research, the datasets used are human-readable, not computer-readable, and they were processed and made suitable for classification using machine learning models. The investigation utilized a CSV file encompassing data pertaining to genes and certain promoters. Before kickstarting, the dataset was constructed in CSV format. Specifically, for dataset details such as available regulators, two datasets were employed: one detailing promoters (including data such as promoter sequence, gene product, transcription unit, strand, start sites, end sites, locus_tag, etc.), and the other describing operons (with details such as genes names, transcription unit, strand, confidence level, etc.).

```
query ($geneName: String!) {
  getOperonBy(search: $geneName) {
    data {
      operon {
        name
      }
      transcriptionUnits {
        name
        promoter {
          name
          confidenceLevel
          regulatorBindingSites {
            function
          }
          sequence  # Assuming this
        }
      }
    }
  }
}
"""
```

Figure 4.2. 1. Query of data of the online RegulonDB.

Considering the impracticality of manually verifying corresponding genes for promoters due to size constraints, Boyer-Moore-Horspool's algorithm was employed to generate a dataset enriched with essential details such as genes, gene names, gene products, promoter names, and regulators. These elements were instrumental in facilitating predictions within gene circuits.

The Hoorspol's algorithm works like show in the below figure:

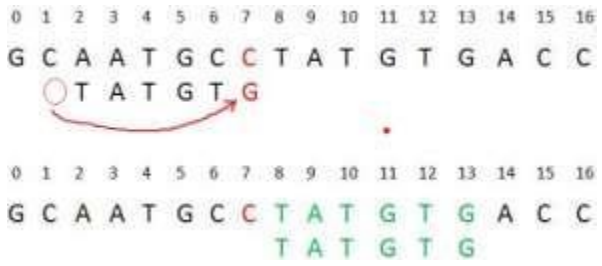

Figure 4.2. 2. Hoorspol's algorithm application

The algorithm operates as depicted in the subsequent figure, applied for both gene and promoter sequences to uncover relationships enabling the identification of promoter regulators.

4.3. Testing

Subsequently, the support vector machine algorithm was implemented to assess the precision with which the algorithm could recognize genes and their characteristics. The gene sequence was transformed into standard features, converting categorical labels into numerical

values, to be recognized by the 'RandomForestClassifier'. Features were standardized using 'StandardScaler', which involved removing the mean and scaling to unit variance. The resulting outcome informed the decision regarding the utilization of gene sequence and promoter sequence for predictions or alternative indicators available in the database.

	precision	recall	f1-score	support
gene	0.57	0.80	0.67	5
operon	0.00	0.00	0.00	3
promoter	0.00	0.00	0.00	1
regulator	0.50	1.00	0.67	1
transcription_unit	0.80	1.00	0.89	4
accuracy			0.64	14
macro avg	0.37	0.56	0.44	14
weighted avg	0.47	0.64	0.54	14

Figure 4. 2. Prediction accuracy of elements linked to the 'gene'.

Based on this data, analysis indicated that gene sequence coding would not be as effective as anticipated. However, an alternative approach ‘transcription_unit’ was considered due to its high precision in identification. The ‘Transcription_unit’ was subsequently employed to identify genes and their corresponding operons. Given that an operon encompasses multiple genes, it may possess distinct promoters. The use of ‘transcription_unit’ facilitated efficient gene and operon identification. A neural network algorithm was then applied to successfully predict potential promoters compatible with these genes within a circuit and their potential functions.

The neural network algorithm code was executed on both databases for promoter classification and prediction based on operon details. The following data representation was utilized.

	operon_name	gene_name	promoter_name	confidence_level	\
0	rpsLG-fusA-tufA	tufA	tufAp1	S	
1	rpsLG-fusA-tufA	tufA	tufAp2	S	
2	rpsLG-fusA-tufA	tufA	fusAp	S	
3	rpsLG-fusA-tufA	fusA	fusAp	S	
4	rpsLG-fusA-tufA	tufA	rpslp	S	

	regulator_functions	promoter_seq
0	[]	tctgcacttcggttcttaccatgacgttgactcctctgaactggcg...
1	[]	tctgcacttcggttcttaccatgacgttgactcctctgaactggcg...
2	[]	tctgcacttcggttcttaccatgacgttgactcctctgaactggcg...
3	[]	gataaatccatggctctgcgcctggcgaacgaactttctgatgctg...
4	[repressor]	tctgcacttcggttcttaccatgacgttgactcctctgaactggcg...

Figure 4. 3. Data representation of a given gene after running 'randomForestClassifier'.

Given the observation that an operon can simultaneously contain multiple promoters and genes, an investigation was conducted to determine the probabilistic distribution between genes and

promoters using the 'RandomForestClassifier' model. This examination sought to explore the likelihood that a single gene could be associated with various promoters or if a single promoter could be associated with various genes, each potentially performing distinct functions. The calculated probability was determined to be **(4,1),** indicating that for each gene, there exists a possibility of matching it with four different promoters.

4.4. Accuracy measure

The most important part is being able to make a prediction with high accuracy; a prediction with low accuracy would not be any different from existing methods. The accuracy of the model is calculated using the formula,

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN)$$

Where TP (true (True Positive) denotes the number of promoter instances (positive) that are correctly classified as promoters, TN (true negative) denotes the number of non-promoter instances (negative) that were correctly classified as negative, FP (false positive) gives the number of negative instances wrongly classified as positive, and FN (false (False Negative) denotes the number of promoters that were incorrectly classified as negative.[18] These parameters are used to calculate various performance metrics like recall score, also known as sensitivity, which is the fraction of positive instances predicted correctly, precision, which is the fraction of predicted positive instances that are actually positive, and f1 score, which is the harmonic mean of recall and precision and the specificity of the model. The study just focused on getting the accuracy number.

The ROC-AUC score measures the area under the Receiver Operating Characteristic curve, which plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings [18]. Mathematically, the AUC is computed as:

$$\text{AUC} = \int_0^1 \text{TPR}(t)\, d(\text{FPR}(t))$$

Using the support vector machine, the accuracy and the ROC-AUC were calculated, where these projected probabilities are accurate.

```
Accuracy: 0.7047353760445683
ROC-AUC: 0.7367907839668637
```

Figure 4. 4. Accuracy of model prediction on overall genes.

4.5. Results

4.5.1. Promoter prediction

Based on these results, an attempt was made to identify a relational connection between data within the given databases. This was undertaken to inform the selection of methodologies suitable for achieving the desired output or for assessing the feasibility of visualizing the transcription unit's role in gene acquisition. The K-Means clustering algorithm was applied, utilizing the transcription unit and operon as central points from which to establish various groupings and connections.

	Source	Source Type	Target	Target Type	Relation
0	C0293	operon	C0293	transcription_unit	contains
1	C0293	transcription_unit	C0293p	promoter	regulated_by
2	C0293	transcription_unit	C0293	gene	contains
3	CsiR	operon	CsiR	transcription_unit	contains
4	CsiR	transcription_unit	glaR	gene	contains
5	aaeR	operon	aaeR	transcription_unit	contains
6	AaeR	transcription_unit	aaeRp	promoter	regulated_by
7	aaeR	transcription_unit	aaeR	gene	contains
8	aaeRp	promoter	repressor	regulator	binds_to
9	aaeXAB	operon	aaeXAB	transcription_unit	contains
10	aaeXAB	transcription_unit	aaeXp	promoter	regulated_by
11	aaeXAB	transcription_unit	aaeX	gene	contains
12	aaeXAB	transcription_unit	aaeA	gene	contains
13	aaeXAB	transcription_unit	aaeB	gene	contains
14	aaeXp	promoter	activator	regulator	binds_to
15	aas-lplT	operon	aas-lplT	transcription_unit	contains
16	aas-lplT	transcription_unit	aas	gene	contains
17	aas-lplT	transcription_unit	lplT	gene	contains
18	aat	operon	aat	transcription_unit	contains
19	aat	transcription_unit	aat	gene	contains
20	abgABT-ogt	operon	abgABT-ogt	transcription_unit	contains
21	abgABT-ogt	operon	ogt	transcription_unit	contains
22	abgABT-ogt	transcription_unit	abgA	gene	contains
23	abgABT-ogt	transcription_unit	abgB	gene	contains
24	abgABT-ogt	transcription_unit	abgT	gene	contains
25	abgABT-ogt	transcription_unit	ogt	gene	contains
26	ogt	transcription_unit	ogtp	promoter	regulated_by
27	ogt	transcription_unit	ogt	gene	contains
28	ogtp	promoter	repressor	regulator	binds_to
29	ogtp	promoter	activator	regulator	binds_to
30	abgR	operon	abgR	transcription_unit	contains
31	abgR	transcription_unit	abgR	gene	contains
32	abpAB	operon	abpAB	transcription_unit	contains
33	abpAB	transcription_unit	abpA	gene	contains
34	abpAB	transcription_unit	abpB	gene	contains
35	abrB	operon	abrB	transcription_unit	contains
36	abrB	transcription_unit	abrB	gene	contains

Figure 4.6. 1. The result of relation of elements between 10 genes, using 'transcription unit' as center.

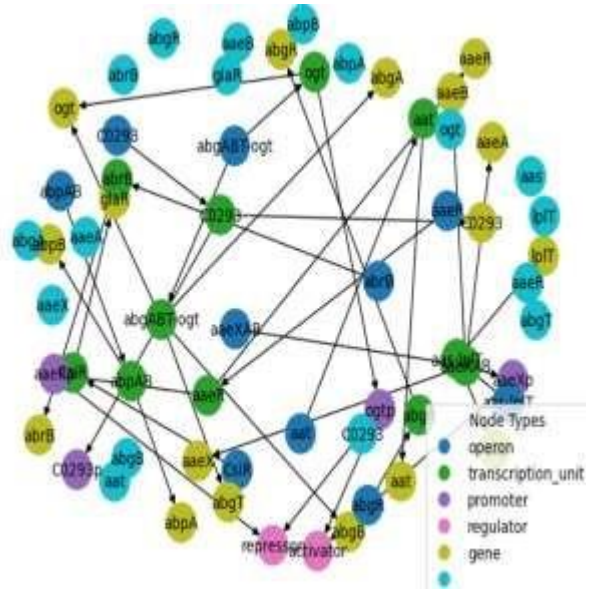

Figure 4.6. 2. The graph shows the connection between 10 genes, regulator, transcription unit and promoter.

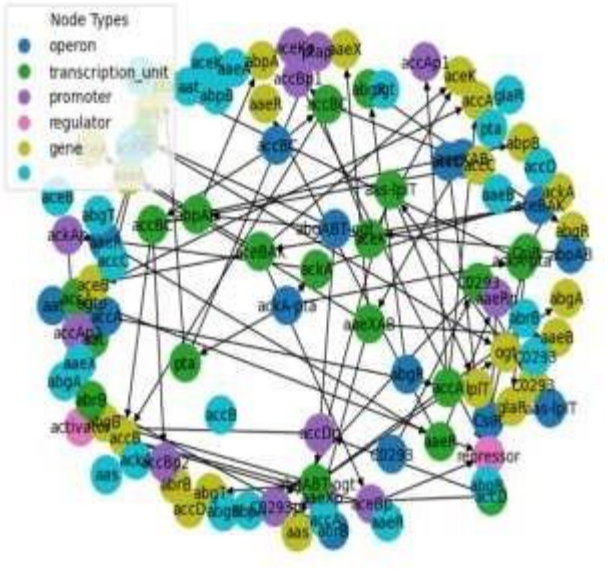

Figure 4.6. 3. The graph shows the connection between 20 genes, regulator, transcription unit and promoter.

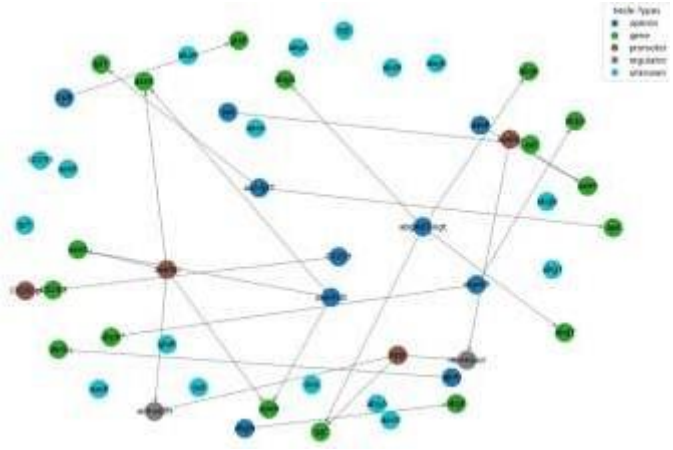

Figure 4.6. 4. The graph shows the connection between gene, regulator and promoter without the 'transcription unit'

This figure, limited to 10 operons within the database, served to validate the role of the 'transcription unit' in facilitating predictions. Observation from the graphical representation indicated numerous connections between the transcription unit, depicted in 'green', and other elements.

Further, the study explored the prediction outcome when provided solely with the gene name. Specifically, the '*FusA'* gene was examined to illustrate the subsequent promoter mapping or recommendations. Following the entry of the gene name, the ensuing results were observed.

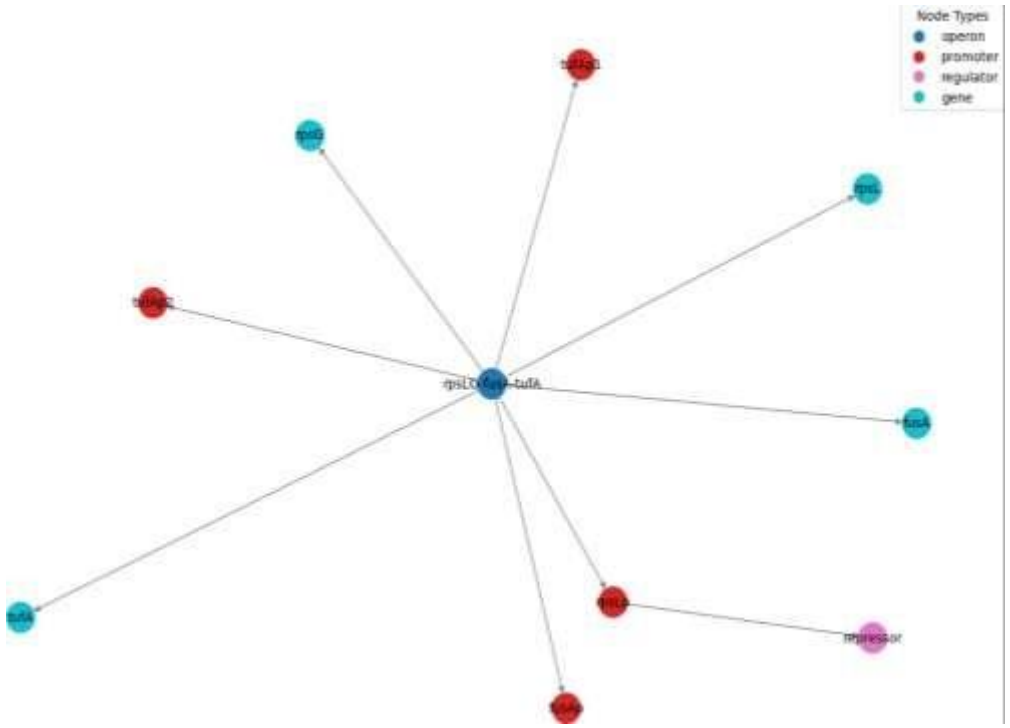

Figure 4.7. 1. This graph shows the predicted details associated with the 'fusA' gene.

```
Please enter the gene name: fusA
       operon_name gene_name                          gene_product
0  rpsLG-fusA-tufA      tufA  translation elongation factor Tu 1
1  rpsLG-fusA-tufA      tufA  translation elongation factor Tu 1
2  rpsLG-fusA-tufA      tufA  translation elongation factor Tu 1
3  rpsLG-fusA-tufA      fusA                 elongation factor G
4  rpsLG-fusA-tufA      tufA  translation elongation factor Tu 1

  promoter_name confidence_level regulator_functions  \
0         tufAp1                S                  []
1         tufAp2                S                  []
2          fusAp                S                  []
3          fusAp                S                  []
4          rpslp                S         [repressor]

                                          promoter_seq
0  tctgcacttcggttcttaccatgacgttgactcctctgaactggcg...
1  tctgcacttcggttcttaccatgacgttgactcctctgaactggcg...
2  tctgcacttcggttcttaccatgacgttgactcctctgaactggcg...
3  gataaatccatggctctgcgcctggcgaacgaactttctgatgctg...
4  tctgcacttcggttcttaccatgacgttgactcctctgaactggcg...
```

Figure 4.7. 2.This image shows the predicted details associated with the 'fusA' gene. It belongs to the 'rpsLG-fusA-tufA' operon and can be linked to the different promoters like: adap2, adap or alkBp with some of their regulator-functions outlined.

Another example is with the *'ada'* gene, with the result suggesting the following:

```
Please enter the gene name: ada
  operon_name gene_name                                gene_product
0    ada-alkB      alkB                    DNA oxidative demethylase
1    ada-alkB       ada  DNA-binding transcriptional dual regulator/DNA...
2    ada-alkB      alkB                    DNA oxidative demethylase
3    ada-alkB       ada  DNA-binding transcriptional dual regulator/DNA...
4    ada-alkB      alkB                    DNA oxidative demethylase

  promoter_name confidence_level    regulator_functions  \
0         adap2                S  [repressor, activator]
1         adap2                S  [repressor, activator]
2          adap                S  [activator, repressor]
3          adap                S  [activator, repressor]
4         alkBp                W                    []
```

Figure 4.8. 1. The graph shows the predicted details associated with the 'ada' gene. It belongs to the 'ada-alkB' operon and can be linked to the different promoters like: adap2, adap or alkBp with its regulator-functions outlined.

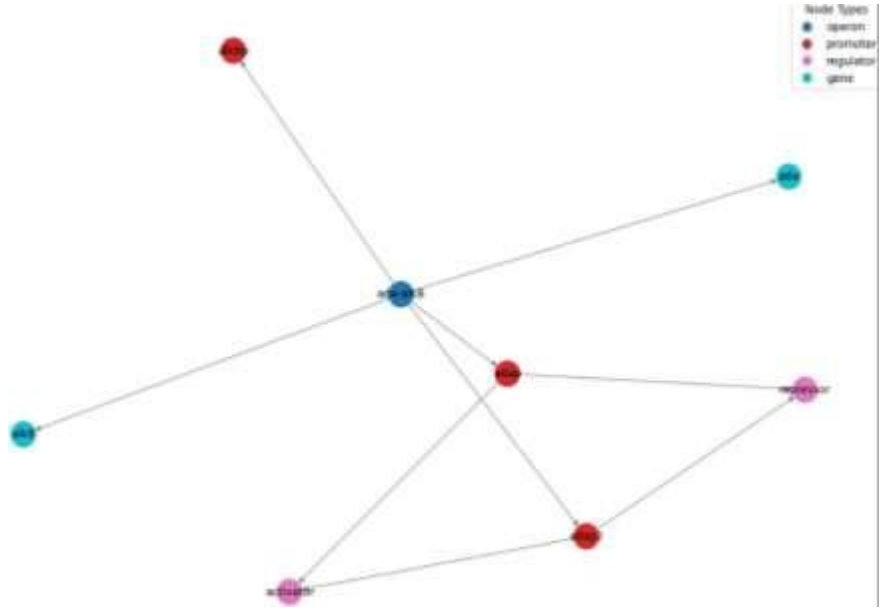

Figure 4.8. 2. This graph shows the predicted elements: Promoter and regulator, associated with the 'ada' gene.

The testing and programming approach did not utilize the function or the regulator; consequently, it did not yield the desired outcome. The goal was to determine which promoter could fulfill requirements when considering functions such as transcription, binding, elongation, and others. The initial results presented all potential solutions rather than a specific one, necessitating further investigation to identify a viable approach.

The results above indicate the operon containing the gene, all genes within that operon, the gene product, which can also be referred to as gene_product, the promoter's name, the confidence level, and the regulator_functions. The confidence level associated with each promoter was of particular interest. This metric reflects **the strength of evidence** supporting the regulatory interaction between a transcription factor (TF) and the target gene controlled by the promoter. Essentially, it quantifies the reliability of the information regarding the promoter's regulation by a specific TF. This information was used to determine the optimal gene and promoter, resulting in the following best configuration for another example gene tested.

```
best_config = optimize_promoter_gene(df, model, vectorizer)
print(f'Best Configuration: {best_config}')

Best Configuration: {'PromoterSeq': 'attactaatataaataaattttgcttgattcatgcaagcggcattaatactatttatactAacgtcaatatacaacccacc',
```

Figure 4.9. 1. A caption of best configuration of 'gspC' gene, indicating the most appropriate promoter sequence that could be used.

Figure 4.9. 2. Continuation of figure 4.9.1's result, with the prediction efficacy value.

‘gspC’ belongs to the operon ‘gspCDEFGHIJKLMO’ and has three optionals promoters:

Table 4. 1. Corresponding genetic elements of the 'gspC' gene.

Gene Name	Operon Name	Promoters	Confidence Level
gspC	gspCDEFGHIJKLMO	gspCp3	Strong (s)
		gspCp2	Weak(w)
		gspCp1	Strong(s)

The best configuration was predicted based on the confidence level of the promoter and the transcription factor, with **an accuracy of 92%** (see in figure 4.9.b), making the optimization factor able to be considered. The promoter sequence "attataata..." in figure 4.9.a suggests the *'gspCp1'* gene, associated with a **strong confidence level**. This implies that a higher confidence level could potentially indicate a more appropriate promoter for the indicated gene.

4.5.2. Regulator classification

The study successfully identified potential promoters for a target gene. However, the primary objective of optimizing promoter selection for a specific gene function was not fully realized due to limitations in current understanding of synthetic biology components and their relationships to gene products. Although, a test was conducted with the genetic algorithm on the available gene products to classify them into functions and determine how effective it could be done.

A genetic algorithm was employed to classify available gene products based on function and derive corresponding theoretical frameworks. Subsequently, a Support Vector Machine (SVM) was utilized to categorize regulator functions into distinct groups. For example, promoters with regulator functions such as DNA_binding, GMP-binding, stpA-binding, etc. were classifier under the 'binding' group, while those with functions like cold shock protein, family protein, periplasmic protein, production protein and more were categorized under the 'protein' group. The classification seemed to work with a rather satisfactory accuracy and cross validation values as shown in figure 5.1.

	precision	recall	f1-score	support
binding	1.00	1.00	1.00	535
other	0.95	1.00	0.98	570
protein	1.00	0.96	0.98	564
regulator	1.00	0.99	0.99	561
transcription	1.00	1.00	1.00	544
accuracy			0.99	2774
macro avg	0.99	0.99	0.99	2774
weighted avg	0.99	0.99	0.99	2774

```
Cross-validation scores: [0.97836668 0.99242834 0.99459167 0.996755   0.98864251]
Average cross-validation score: 0.9901568415359655
```

Figure 4.10. 1. Value of precision, recall, f1-score, cross validation and average cross validation of the regulator functions classification using Support Vector Machine.

Figure 5.1 demonstrates that the algorithm accurately predicted the following gene for the 'binding' functionality with 98% precision. Nevertheless, predictive accuracy for the remaining gene products was inadequate.

```
Genes for binding: ['acrR', 'ada', 'adiY', 'ageR', 'alaS', 'allR', 'allS', 'alsB', 'alsR', 'alsR', 'appY', 'appY', 'araC', 'araF', 'araF',
```

Figure 4.10. 2. List of genes that could be used for 'binding' functionality.

Additional predictions for 'transcription' and 'protein' functions were conducted, with results summarized in Figure 5.3. Accuracy for these predictions ranged from 87% to 92%.

```
Genes for transcription: ['aaeR', 'aaeR', 'aaeR', 'aaeR', 'aaeR', 'acrR', 'ada', 'adiY',
Genes for protein: ['aaeX', 'aat', 'accB', 'accB', 'acnB', 'acnB', 'acpP', 'acrA', 'acrE
```

Figure 4.10. 3. List of genes that could be used for 'transcription' or as 'proteins'

Chapter 5: Conclusions and Recommendation

5.1. Validation

Validation was conducted by identifying studies that experimentally investigated specific operons and determining if the predicted promoter matched the one used. In the present case (figure 4.7.1), '**FusA'** was associated with the '**rpsLp'** promoter, along with other genes such as '**Tuf'**, with high confidence. Tieleman et al. demonstrated growth phase-dependent transcription of the *Streptomyces ramocissimus* (tuf) gene from two promoters, confirming successful promoter function in laboratory conditions. [19]

A separate study explored the relationship between '**ADA'** gene mutations and severe combined immunodeficiency (SCID), as well as available treatment options. Gene therapy was presented as a potential treatment, with a detailed description of an initial trial involving cells transduced with the **'ADA'** gene. The study employed the '. ttgt.' promoter, termed 'adap', to initiate gene transcription. Notably, this promoter aligns with the promoter predicted (see figure 4.8.2) in the current experiments with strong confidence. [20]

5.2. Discussion

Based on the results obtained, the research questions may be potentially answered.

1. How can ML algorithms be tailored to accurately predict promoter efficacy in driving specific gene expression patterns?

Tailoring machine learning (ML) algorithms to accurately predict promoter efficacy in driving specific gene expression patterns involves a multifaceted approach that integrates various data sources beyond mere sequence information. This includes leveraging data on transcription factor binding sites, operon structures, and gene expression patterns across different conditions to capture the complexity of gene regulation. Exploring deep learning architectures, such as convolutional neural networks (CNNs), can enable the learning of intricate relationships between promoter sequences and gene expression profiles. Our research focused on applying ML algorithms, specifically support vector.

tor machines and random forest classifiers, to predict promoter efficacy. Through preprocessing gene and promoter datasets and utilizing algorithms like Boyre More Horspool's for constructing relationships between gene sequences and their promoters, we discovered that while direct gene sequence coding was less effective, incorporating transcription unit data significantly enhanced prediction accuracy. This strategy demonstrates the value of including biologically relevant features, such as transcription units, in ML models to achieve higher accuracy in predicting promoter behaviors that influence specific gene-expression patterns.

2. What computational strategies can be employed to sift through the myriads of possible promoter-gene configurations to identify those most conducive to desired genetic circuit outcomes?

Incorporating advanced computational strategies and domain knowledge from synthetic biology can significantly enhance the efficiency and effectiveness of identifying optimal promoter-gene configurations for desired genetic circuit outcomes. By leveraging established genetic circuit designs and expression data from successful synthetic biology projects, researchers can guide the search for the most promising promoter-gene combinations. Swarm intelligence algorithms, such as particle swarm optimization, offer a valuable tool for navigating the extensive configuration space, enabling efficient exploration of potential solutions. Clustering algorithms, particularly K-means, have been effectively utilized to simplify the complexity of identifying optimal configurations by grouping promoters and genes based on their associations with transcription units and operons. Visualization of these clusters reveals patterns that highlight which promoters are most likely to interact with specific genes within operons, facilitating the selection of configurations that are most conducive to achieving desired genetic circuit outcomes. Further insights into the diversity of genetic circuit designs can be gained by analyzing probabilistic distributions using models like the random forest classifier, which helps understand the likelihood of multiple promoters interacting with a single gene.

5.3 Limitations and Recommendations

5.3.1. Limitations

We did not design an optimization algorithm that identifies the best promoter-gene configurations for achieving specified synthetic biology applications. With our current research testing and results, it is very challenging to definitely call it a success. We explore the possibility, and we believe it is possible, with a strong knowledge of genes and their different applications and functions, as well as finding the right data. Other limitations include:

- Data Quality and Quantity: The study faced challenges due to its reliance on datasets lacking functional annotations for gene products, limiting the model's performance enhancement potential.
- Single Functionality Focus: The current model is designed to identify promoters for a singular predicted function, restricting its applicability to scenarios requiring promoters for multiple functionalities.
- Machine Learning Reliance: There was a noticeable emphasis on algorithmic approaches over the development and training of machine learning models, suggesting a missed opportunity to leverage supervised learning techniques for improved prediction accuracy. Despite the anticipated dominance of ML, the project underscored the critical role of computational algorithms over traditional machine-trained algorithms, highlighting the challenges posed by dataset quality and the necessity of matching promoters to genes without strict binding requirements.
- Limited Generalizability: Validation primarily depended on a single source, reducing the overall generalizability of the findings. Expanding validation methods to include a broader array of experimental data could enhance the study's applicability.

A stronger foundation and deep understanding in Synthetic Biology, particularly in the study and understanding of genetic circuits, would have improved the outcome of this work.

5.3.2. Recommendations and Future Studies

A recent discussion on ResearchGate featured a scientist's query as depicted in Figure 5.1. While the desired gene function was clear, the associated elements, such as promoters, remained uncertain. Recommendations and potential outcomes were sought. Further investigation into this proposed approach could facilitate the establishment of these connections and enable the development of a user application capable of providing automated gene details and applications in response to user inquiries about gene functions.

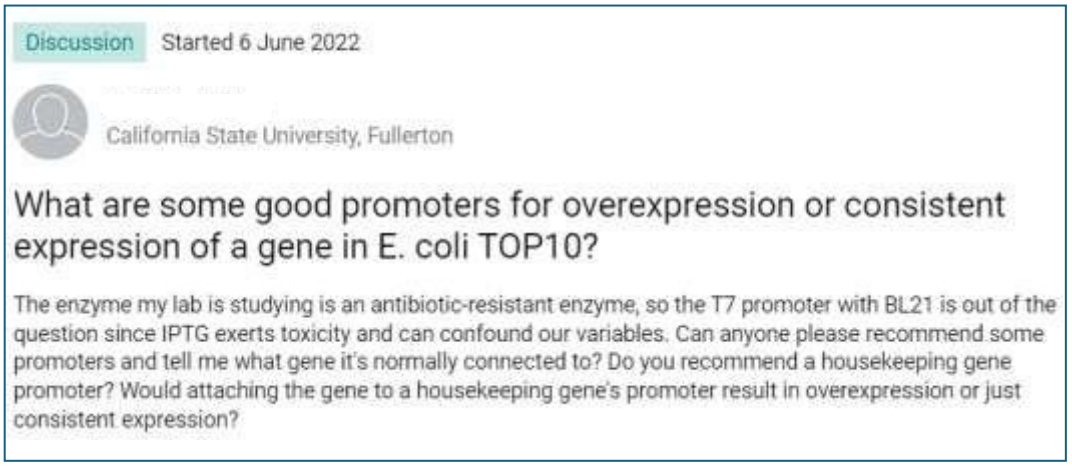

Figure 5. 1. Question asked by a synthetic biologist on research gate. (Name of person covered)

The study could also improve based on the given recommendations:

- Dataset Expansion: Prioritize the acquisition and integration of datasets enriched with functional annotations for gene products and validated promoter-gene interactions to bolster model performance.
- Machine Learning Techniques: Investigate and implement supervised machine learning algorithms for promoter prediction, taking advantage of the expanded datasets to enhance prediction accuracy. To further refine ML algorithms for promoter efficacy prediction, strategies such as feature selection, integration of domain knowledge, data augmentation, and model optimization are recommended.
- Multi-Functionality Prediction: Develop models capable of predicting promoters for multiple functionalities, utilizing gene sequences and operon information to cater to a

broader range of synthetic biology applications. This could be done using genetic algorithms as shown in some of the literature reviewed.

- Rigorous Validation: Adopt more comprehensive validation methods, encompassing a variety of experimental data and established benchmark datasets, to ensure the robustness, efficacity and generalizability of the findings.
- Collaboration: Foster collaborations between synthetic biologists and computer scientists to deepen the understanding of gene function and its correlation with promoter selection, thereby informing feature selection and interpretation in ML model development.

5.4. Conclusion

Machine learning-based systems have the potential to revolutionize the field of synthetic biology. This includes the creation of novel promoter sequences for both existing and hypothetical genes, as demonstrated by the application of genetic algorithms in this research. The findings suggest promising avenues for the automation of design and testing processes with increased confidence in successful outcomes.

Future research should focus on developing more comprehensive databases of specific organisms and integrating diverse knowledge sources to optimize algorithms for identifying optimal promoter configurations for specific functions. Additionally, the implementation of computer algorithms capable of understanding the structure of biological sequences would simplify the complex task of relational analysis.

References

[1] Mouawad, C. 2020. The origin and history of synthetic biology. BCC Research. Retrieved February 10, 2024, from https://blog.bccresearch.com/the-origin-and-history-of-synthetic-biology

[2] Hiscock, T.W. 2019. Adapting machine-learning algorithms to design gene circuits. BMC Bioinformatics, 20(1). https://doi.org/10.1186/s12859-019-2788-3

[3] Brophy, J.A.N., & Voigt, C.A. 2014. Principles of genetic circuit design. National Institutes of Health. https://doi.org/10.1038/nmeth.2926

[4] Kim, J., Salvador, M., Saunders, E., González, J., Avignone-Rossa, C., & Jiménez, J. I. (2016). Properties of alternative microbial hosts used in synthetic biology: towards the design of a modular chassis. Essays in Biochemistry, 60(4), 303–313. https://doi.org/10.1042/ebc20160015

[5] Jbei. 2022. Using machine learning and synthetic biology to combat climate change. JBei.org. https://www.jbei.org/using-machine-learning-and-synthetic-biology-to-combat-climateChange

[6] Nandagopal, N., & Elowitz, M. B. (2011). Synthetic Biology: Integrated Gene Circuits. *Science*, 333(6047), 1244–1248. DOI: 10.1126/science.120708

[7] Xiang, Y., Dalchau, N., & Wang, B. 2018. Scaling up genetic circuit design for cellular computing: advances and prospects. Natural Computing, 17(4), 833-853. https://doi.org/10.1007/s11047-018-9715-9

[8] Lucks, J., Qi, L., Whitaker, W., & Arkin, A. 2008. Toward scalable parts families for predictable design of biological circuits. Current Opinion in Microbiology, 11(6), 567-573. https://doi.org/10.1016/j.mib.2008.10.002

[9] Tang, H., Wu, Y., Deng, J., Chen, N., Zheng, Z., Wei, Y., Luo, X., & Keasling, J. D. (2020). Promoter Architecture and Promoter Engineering in Saccharomyces cerevisiae. *Metabolites*, 10(8), 320. Retrieved from https://doi.org/10.3390/metabo10080320

[10] Kotopka, B. J., & Smolke, C. D. (2019). Production of the cyanogenic glycoside dhurrin in yeast. *Metabolic Engineering Communications*, *9*, e00092. https://doi.org/10.1016/j.mec.2019.e00092

[11] Zhu, J., Zhang, Q., Forouraghi, B., & Wang, X. 2021. Applications of machine learning techniques in genetic circuit design. ACM. https://doi.org/10.1145/3457682.3457683

[12] Saltepe, B., Bozkurt, E.U., Güngen, M.A., Çiçek, A.E., & Şeker, U.Ö.Ş. 2021. Genetic circuits combined with machine learning provides fast responding living sensors. Biosensors and Bioelectronics, 178, 113028. https://doi.org/10.1016/j.bios.2021.113028

[13] Aromolaran, O., Aromolaran, D., Isewon, I., & Oyelade, J. 2021. Machine learning approach to gene essentiality prediction: a review. Briefings in Bioinformatics. https://doi.org/10.1093/bib/bbab128

[14] Dixit, P., & Prajapati, G. 2015. Machine Learning in Bioinformatics: A Novel Approach for DNA Sequencing. 2015 Fifth International Conference on Advanced Computing & Communication Technologies, 41-47. https://doi.org/10.1109/ACCT.2015.73

[15] Vahid, A. M., Ahmad, M., and Frekri, A. (2015). "Promoter Prediction in Bacterial DNA Sequences Using Expectation Maximization and Support Vector Machine Learning Approach." *Journal of Data Mining in Genomics & Proteomics*, 6(2).

[16] Bhasin, H., & Mehta, S. 2015. On the applicability of diploid genetic algorithms. AI & SOCIETY, 31, 265-274. https://doi.org/10.1007/s00146-015-0591-x

[17] Santoso, W., Hulliyah, K., Nurjannah, W., & Setianingrum, A. 2022. *Systematic Literature Review: Virus Prediction Based on DNA Sequences using Machine Learning and Deep Learning method*. Retrieved February 10, 2024, from https://www.semanticscholar.org/paper/Systematic-Literature-Review%3A-Virus-Prediction-on-Santoso-Hulliyah/26b848bc0e8c2b84def343840c4bee7424f6b7a2.

[18] Sundareswaran, A., Aathavan, B., & Jaisankar, N. 2019. Promoter prediction in DNA sequences of Escherichia coli using machine learning algorithms. ResearchGate. https://www.researchgate.net/publication/343318555_Promoter_Prediction_in_DNA_Sequences_of_Escherichia_Coli_Using_Machine_Learning_Algorithms

[19] Olsthoorn-Tieleman, L. N., Fischer, S. E. J., & Kraal, B. (2002). The Unique tuf2 Gene from the Kirromycin Producer Streptomyces ramocissimus Encodes a Minor and Kirromycin-Sensitive Elongation Factor Tu. *Journal of Bacteriology*, 184(15), 4211--4218. https://doi.org/10.1128/jb.184.15.4211-4218.2002

[20] Muul, L. M. (2002). Persistence and expression of the adenosine deaminase gene for 12 years and immune reaction to gene transfer components: long-term results of the first clinical gene therapy trial. *Blood*, *101*(7), 2563–2569. https://doi.org/10.1182/blood-2002-09-2800

[21] Cameron, D.E., Bashor, C.J., & Collins, J.J. 2018. A method for measuring the properties of a genetic circuit module in the presence of resource sharing. In 2018 IEEE Conference on Decision and Control (CDC) (pp. 4299-4304). IEEE.

[22] Pandi, A., Diehl, C., Kharrazi, A. Y., Scholz, S. A., Bobkova, E., Faure, L., Nattermann, M., Adam, D., Chapin, N., Foroughijabbari, Y., Moritz, C., Paczia, N., Cortina, N. S., Faulon, J., and Erb, T. J. 2022. A versatile active learning workflow for optimization of genetic and metabolic networks. *Nature Communications*, 13(1). DOI: 10.1038/s41467-022-31245-z.

[23] Schroeder, W.L., Baber, A.S., & Saha, R. 2021. Optimization-based Eukaryotic Genetic Circuit Design (EuGeneCiD) and modeling (EuGeneCiM) tools: Computational approach to synthetic biology. iScience, 24(9), 103000. https://doi.org/10.1016/j.isci.2021.103000

[24] Munch, R. 2003. PRODORIC: prokaryotic database of gene regulation. Nucleic Acids Research, 31(1), 266–269. https://doi.org/10.1093/nar/gkg037

[25] Ireland, W.T., Beeler, S.M., Flores-Bautista, E., McCarty, N.S., Röschinger, T., Belliveau, N.M., Sweredoski, M.J., Moradian, A., Kinney, J.B., & Phillips, R. 2020. Deciphering the

regulatory genome of Escherichia coli, one hundred promoters at a time. eLife, 9. https://doi.org/10.7554/elife.55308

[26] Chakraborty, D., Rengaswamy, R., & Raman, K. 2021. Designing biological circuits: from principles to applications. ACS Synthetic Biology. https://doi.org/10.1021/acssynbio.1c00557

[27] Wang, Y., Tai, S., Zhang, S., Sheng, N., & Xie, X. 2023. ProMGER: Promoter prediction based on graph embedding and ensemble learning for eukaryotic sequence. Genes, 14(7), 1441. https://doi.org/10.3390/genes14071441

[28] Wu, D., Karhade, D., Pillai, M., Jiang, M., Huang, L., Li, G., Cho, H., Roach, J., Li, Y., & Divaris, K. 2021. Machine Learning and Deep Learning in Genetics and Genomics. Machine Learning in Dentistry. https://doi.org/10.1007/978-3-030-71881-7_13

[29] Hershberg, R.H., Bejerano, G.B., Santos-Zavaleta, A.S., & Margalit, H.M. 2001. PromEC: An updated database of Escherichia coli mRNA promoters with experimentally identified transcriptional start sites. NCBI. Retrieved June 1, 2024, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC29777/

[30] Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., & Chouvarda, I. 2017. Machine Learning and Data Mining Methods in Diabetes Research. Computational and Structural Biotechnology Journal, 15, 104-116. https://doi.org/10.1016/j.csbj.2016.12.005

Glossary

Synthetic Biology: an interdisciplinary field that combines biology, engineering, and computer science to design and construct new biological systems, organisms, or biological processes.

Genetic circuit: it refers to engineered biological pathways composed of genetic components, such as DNA sequences, promoters, and gene regulatory elements.

DNA (Deoxyribonucleic acid): a molecule that contains the genetic instructions used in the development and function of all living organisms.

Escherichia Coli: a type of Gram-negative, rod-shaped bacterium that is commonly found in the lower intestines of warm-blooded animals. It is served as a model organism for studying bacterial genetics, metabolism, and pathogenesis.

Promoter: a region of DNA that serves as a binding site for RNA polymerase, the enzyme responsible for initiating transcription in prokaryotes and eukaryotes.

Operon: a genetic regulatory unit found in bacteria and archaea, consisting of a cluster of genes that are transcribed together as a single unit.

Gene: A unit of heredity and a segment of DNA that contains the instructions for the development and function of an organism.

Regulators: the role of a gene or its product in controlling the expression of other genes. They can act as transcription factors, enhancers, or silencers, influencing the transcription of target genes.

Protein: enabler of individual cells to respond and interact with each other to perform some logical functions.

Transcription Unit: sequence of nucleotides in DNA that codes for a single RNA molecule, along with the sequences necessary for its transcription; normally contains a promoter, an RNA-coding sequence, and a terminator.

Locus Tag: identifiers that are systematically applied to every gene in a genome.

Strand: promoter sequence defines the direction of transcription and indicate which DNA strand will be transcribed.

Gene product: the result of the expression of a gene, which is typically a protein or RNA molecule.

Promoter or gene sequence: a promoter is a specific DNA sequence that serves as a signal.