Optimization of Artificial Operon Construction by Consultation Algorithms Utilizing LCS

Changhee Han[1], Kenji Tsuge[2], and Hitoshi Iba[1]
[1]Graduate School of Information Science and Technology, The University of Tokyo, Bunkyo, Tokyo, Japan
[2]Institute for Advanced Biosciences, Keio University, Tsuruoka, Yamagata, Japan
Email: maikallis@yahoo.co.jp, ktsuge@ttck.keio.ac.jp, iba@iba.t.u-tokyo.ac.jp

***Abstract*—How can we boost *Escherichia coli* (*E. coli*) growth (i.e., production) without modifying genes themselves? This remains a challenging and fruitful goal that would facilitate the mass production of biofuels, biomedicine, and engineered genomes in synthetic biology. In this paper, we focus on rearranging gene order within an operon to optimize gene expression. Optimizing more than five genes remains laborious without predictive modeling as the number of gene orders increases factorially—a five-gene operon possesses 120 gene orders, but a ten-gene operon possesses 3,628,800 gene orders. To handle a ten-gene operon, we propose consultation algorithms utilizing LCS to analyze the relationship between gene order and growth rate, and then verify predicted gene orders with high growth rates using wet-lab experiments. "Consultation" refers to optimizing gene orders in different machine learning algorithms and choosing gene orders with high growth rates in each algorithm to avoid over-fitting. We address the following research questions: (RQ1) How can we predict *E. coli* growth according to gene orders? (RQ2) Can definite rules easily understood by biologists be extracted? (RQ3) Can new *E. coli* strains surpass the highest growth rate of the dataset? Our first computational approach shows that consultation algorithms utilizing LCS can identify gene orders that significantly control *E. coli* growth and create novel *E. coli* strains with high growth rates using these operon construction rules.**

I. Introduction

How can we boost *Escherichia coli* (*E. coli*) growth (i.e., production) without modifying genes themselves? In this paper, we address rearranging gene order within an operon to optimize gene expression by machine learning, which is one of the most important topics in synthetic biology. Synthetic biology [1] has the potential to generate useful materials produced by complex artificial biological systems for applications, such as personalized medicine [2], medical diagnosis [3], and advanced biofuels [4]. Metabolic engineering advances, such as multiple gene assembly technologies, have contributed to progress in this field. The ordered gene assembly in *Bacillus subtilis* (OGAB) method [5] facilitates the assembly of multiple DNA fragments in a fixed order and orientation, thus potentially boosting *E. coli* growth by rearranging gene order within an operon; again, because gene order within an operon influences gene expression, this is a straightforward means of adjusting *E. coli* growth rates.

In the past, synthetic biologists have empirically executed metabolic engineering of biosynthesis pathways, such as those of carotenoids [6], [7] and P(3HB) [8], by optimizing gene order without computational predictive modeling. But optimizing more than five genes remains elusive because the number of gene orders increases factorially.

Our aim was to investigate the influence of gene order among ten genes within an operon on the growth rate of *E. coli*. To do so, we utilized consultation via machine learning algorithms that include learning classifier systems (LCS) [9], [10], [11] to analyze the relationship and then verified predicted gene orders with high growth rates using wet-lab experiments.

Research Questions. In this paper, we mainly address the following three research questions:

- **(RQ1) Prediction:** How can we predict *E. coli* growth according to gene orders?
- **(RQ2) Rule extraction:** Can definite rules easily understood by biologists be extracted?
- **(RQ3) Growth rate:** Can new *E. coli* strains surpass the highest growth rate of the dataset? And if not, what is the highest rate achievable?

Contributions. Our main contributions and findings are summarized as follows:

- **Artificial operon construction:** This is the first computational approach for analyzing the relationship between operon gene order and *E. coli* growth rate for comprehensive prediction. Extracted two-gene order rules can be used to design operons that significantly effect *E. coli* growth. Our study also reveals that surpassing the highest growth rate of the dataset is challenging.
- **Synthetic biology:** This research significantly contributes to the goal of designing efficient operons for the mass-production of useful materials in synthetic biology. Furthermore, this study provides a crucial step toward interdisciplinary research linking LCS and synthetic biology, which can be applied to various other tasks.

Outline. The rest of the paper is structured as follows: Section II provides a brief summary of background information on LCS, synthetic biology, gene expression analysis with LCS, and operon structure optimization. In Section III, the particular artificial operon model is defined. The experimental procedure of both computational simulations and their biological verification is presented in Section IV. Section V reports the results of the experiments. Finally, Section VI presents our findings and concludes the paper with some proposed future developments.

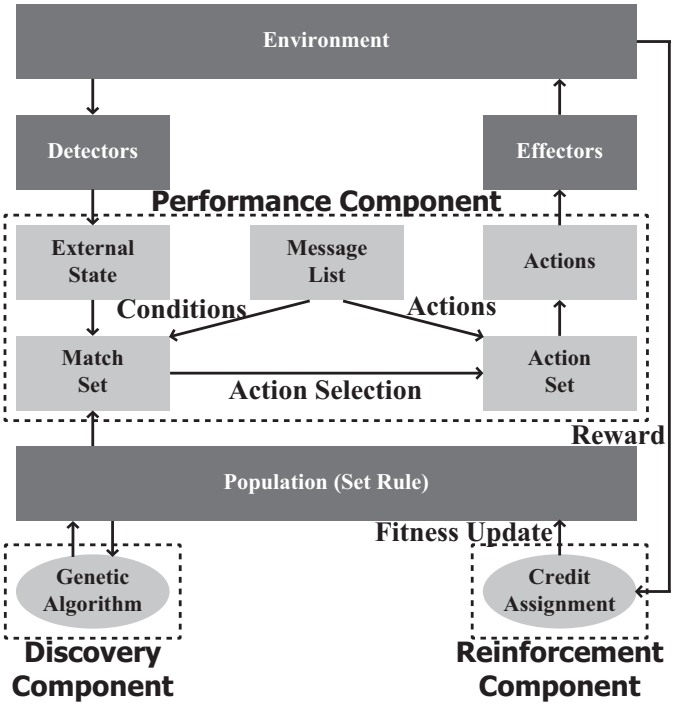

Fig. 1. Interactions among LCS components.

II. Background

A. *Learning Classifier Systems*

LCS are algorithms that incorporate genetic algorithms (GA) with reinforcement learning (RL) to produce adaptive systems described by simple if-then rules. Nature-inspired LCS were originally envisioned as cognitive systems with interactive components (Fig. 1), permitting the classification of massive and noisy datasets in biological systems. Furthermore, LCS efficiently generate compact rules describing complex systems that enable knowledge extraction. Hence, LCS can efficiently extract knowledge from biological system datasets as well as previously intractable dynamic systems, such as affective image classification in spatial-frequency domain [12].

Accordingly, LCS and LCS-inspired systems have solved many bioinformatic and medical problems, such as automating alphabet reduction for protein datasets [13] and the classification of a primary breast cancer dataset [14]. LCS-inspired systems, like BioHEL [15], [16], exist that are even designed for data mining large-scale bioinformatic datasets.

B. *Synthetic Biology*

Engineering microorganisms accurately requires a comprehensive understanding of natural biological systems, such as cryptic cellular behaviors, and tools for controlling cells. To overcome these problems, as a relatively new interdisciplinary branch of biology, synthetic biology aims to design and construct complex artificial biological systems from the bottom-up for practical applications. LCS can be applied to fields ranging from renewable biofuels [17] to biomedicine—for intractable diseases such as cancer [18], infectious diseases [19], and autoimmune disorders [20]—and engineered genomes [21] of artificial organisms.

In synthetic biology, the engineering design cycle for the core platforms exists to avoid laborious trial and error; this design cycle illustrates how to (1) design systems according to high-level concepts, (2) model these designs as circuits with efficient parts libraries, (3) simulate their functionality, (4) construct the design effortlessly, (5) probe the resulting circuits, and (6) measure the results. Phases such as circuit conceptualization, design, and construction have advanced significantly, but many bottlenecks still exist at modeling, simulation, probing, and measurement phases. In the design cycle, constant feedback between stages plays a key role in enhancing circuit functionality. Moreover, evolutionary strategies exist in the cycle to increase the performance of other steps, though these strategies remain underutilized.

There is a rising trend in designing artificial metabolic pathways that show previously undescribed reactions produced by the assembly of enzymes from different sources in a single host. However, few researchers have succeeded thus far because of the difficulty of empirically analyzing gene expression that determines the synthesis of a functional gene product; in synthetic biology, optimizing gene expression remains a challenging and potentially fruitful goal that would facilitate the mass production of useful products such as biofuels. In order to avoid empirical studies without undertaking predictive modeling, synthetic biology is shifting from developing proof-of-concept designs to establishing general core platforms for efficient biological engineering based on computer science [22]. In this context, data mining and knowledge discovery are essential to investigate and utilize natural biological phenomena.

C. *Gene Expression Analysis with LCS*

There is a rapid rise of interest in the application of evolutionary computation techniques in many different domains, including computer science, engineering, and bioinformatics. Recently, Al-Sahaf *et al.* [23] succeeded to utilize a genetic programming based image descriptor for multiclass texture classification with only two instances per class. In the context of bioinformatics, Chen *et al.* [24] proposed multi-dimensional scaling and MODELLER-based evolutionary algorithms for protein model refinement. Furthermore, the multiobjective evolutionary algorithm based on NSGA-II by Ortuno *et al.* [25] optimized multiple sequence alignment. Similarly, synthetic biology is also not the exception. Naruse *et al.* [26] attmpted to create genetic networks using differential evolution and succeeded to extract the knowledge about robust network structures. Likewise, the estimation of parameters of gene regulatory networks was successfully modeled according to the S-system formalism by Nobile *et al.* [27].

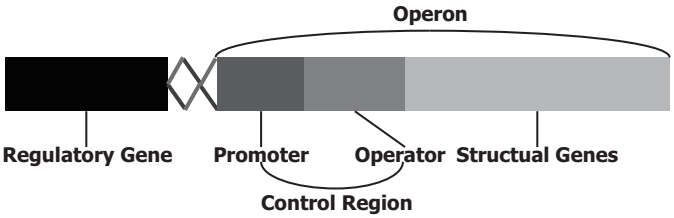

Fig. 2. **Operon structure** consisting of a promoter, an operator, and structural genes, which is controlled by a regulatory gene.

As explained in the previous subsection, gene expression analysis plays an essential role in solving a large number of problems in synthetic biology. LCS have recently been attempted in a wide range of genetics and genomics fields related to the mechanisms of gene expression because of their ability to interpret large and complex genomic datasets. Glaab *et al.* [28] evaluated rule-based evolutionary machine learning systems inspired by Pittsburgh-style LCS, specifically BioHEL, and GAssist, in order to increase the understandability of prediction models with high accuracy. Zibakhsh *et al.* [29] suggested memetic algorithms, including LCS with a multi-view fitness function approach. These algorithms significantly outperformed classic memetic algorithms in discovering rules. Abedini *et al.* [30] proposed two XCS-inspired evolutionary machine learning systems to investigate how to improve classification accuracy using feature quality information: FS-XCS, which utilize feature selection to reduce features, and GRD-XCS, which exploit feature ranking to modify the rule discovery process of XCS. Implementation of LCS to deal with gene expression is poised to grow in the near future, as a number of sizeable genomic datasets are made available by large-scale international projects, such as the 1000 Genomes Project, the 100,000 Genomes Project, and the ENCODE Project.

D. Operon Structure Optimization

A number of genes in genomes encode many proteins that modulate cellular activities or implement specific functionality. In bacterial genomes, such as *E. coli* genomes, such genes frequently act as an operon, that is, a functioning unit of genomic DNA that controls the transcription of multiple genes simultaneously with a single promoter. Fig. 2 illustrates a typical operon with a promoter, an operator, and structural genes. An operon is transcribed into a continuous mRNA strand and either translated in the cytoplasm, or trans-spliced to generate monocistronic mRNAs that are translated independently. As such, gene expression of elements within an operon decrease linearly with transcription distance [31]. To increase the productivity of synthetic metabolic pathways, the relative abundance of transcripts must be regulated accurately using an operon to achieve the balanced expression of multiple genes and avoid the accumulation of toxic intermediates or bottlenecks that inhibit the growth of microorganisms [32].

To approach this central challenge in synthetic biology, operon structure optimization has been pursued in recent years. There are several examples: the optimization and genetic implementation of a blueprint as an artificial operon, according to metabolic real-time analysis, tripled the production of dihydroxyacetone phosphate from glucose [33]; the amplification of genomic segments in artificial operons successfully controlled gene expression using selective RNA processing and stabilization (SRPS) [34], which transcribes primary mRNA into segments using nucleases and thus produces variation in stability among the segments [35]; libraries of tunable intergenic regions (TIGRs)—which recombine numerous post-transcriptional control elements and permit specifying the desired relative expression levels—have helped optimize gene expression in artificial operons [36].

Instead of modifying genes themselves, a completely different approach—reordering multiple genes into an operon structure with an appropriate promoter—may become a breakthrough in optimizing *E. coli* growth (i.e., production).

III. Problem Definition

We applied machine learning to investigate the construction principles relating gene order within an operon to the separation from a promoter involved in a metabolic pathway and thus the growth rate of *E. coli*. The operon contained ten genes (labeled A, B, C, D, E, F, G, H, I, and J). We then verified the high-growth-rate gene orders using wet-lab experiments.

The expression of multiple genes changes in accordance with changes in gene order [32]. Therefore, *E. coli* growth rates differ not only as a result of the presence and absence of genes, but also gene order within operons. Generally, if the expression of multiple genes is balanced, *E. coli* grows rapidly, but if it is not, *E. coli* grows poorly or is totally inhibited as toxic intermediates or bottlenecks accumulate. Yet, optimizing more than five genes has been impossible using conventional approaches because the number of gene orders increases factorially with the number of genes in an operon.

We identified ten genes in this study. It was challenging to analyze them accurately for the following reasons: (1) the number of gene orders obtained from wet-lab experiments was only 0.004% of the total 3,628,800 gene orders (the dataset consists of 93 individual datasets from *E. coli* with gene orders similar to wild-type strains as well as 51 datasets with random gene orders); (2) even *E. coli* with identical gene orders exhibit a large standard deviation in growth rate (the maximum standard deviation of our dataset is around 0.05/h); (3) *E. coli* strains with high growth rates in the dataset possess similar gene orders—the highest growth rate was approximately 0.73/h.

Therefore, we adopted algorithms of heuristic machine learning techniques that can resolve the trade-off between accuracy and smoothness to elucidate which gene orders significantly influence *E. coli* growth. As Fig. 3 shows, gene order refers to the arrangement of genes, and it significantly

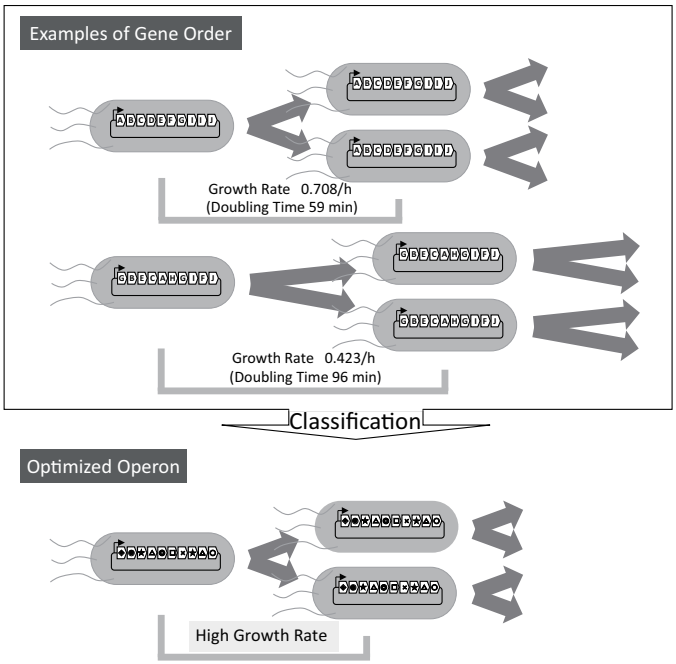

Fig. 3. **Optimizing gene order** of an artificial operon.

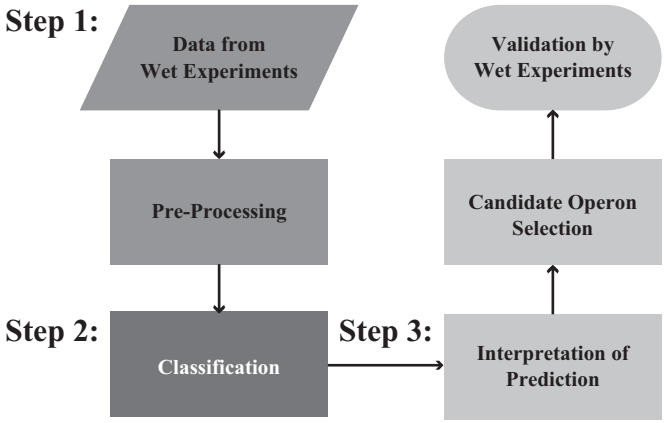

Fig. 4. **Flowchart illustrating the experimental procedure**. The protocol consists of three steps: (1) pre-processing; (2) supervised analysis; and (3) post-analysis.

influences the growth rate of *E. coli*. Accordingly, we investigated operon construction principles, how rearranging gene order within an operon influences gene expression and thus boosting *E. coli* growth, and tried to design new *E. coli* strains with high growth rates.

Fig. 4 provides a flowchart describing the experimental procedure. Throughout the wet-lab experiment, *E. coli* was cultivated in duplicate. The OGAB method was exploited to reconstitute gene orders by assembling multiple DNA fragments with a desired order and orientation and thus generating an operon structure in a resultant plasmid. We normalized the raw data and classified the gene orders into several growth rates; we selected a sampling of gene orders and verified them through wet-lab experiments.

TABLE I
CLASSES OF GROWTH RATES FOR TWO-ALGORITHM CONSULTATION

Classes	Growth Rate	Classes	Growth Rate
0	0–0.1/h	0.1	0.1–0.2/h
0.2	0.2–0.3/h	0.3	0.3–0.4/h
0.4	0.4–0.5/h	0.5	0.5–0.6/h
0.6	0.6–0.7/h	0.7	$\geq$ 0.7/h

IV. EXPERIMENTAL FRAMEWORK

We conducted two experiments with consultation algorithms, one employing the results of two algorithms—Pittsburgh-style LCS [9], [10], [11] called GAssist [37] for compact solutions with few rules and a decision-tree induction technique, C5.0—and the other employing the results of four algorithms—other two well-known conventional machine learning techniques, random forest and multilayer perceptron, in addition to using LCS and C5.0. This was executed in order to analyze the relationship between the gene order within operons and *E. coli* growth rate. Then, we designed six operons per experiment according to the optimal predicted gene orders to assess the performance of the classifiers. Particularly, six operons per experiments were selected because of the difficulty of designing a large number of operons in terms of time and effort.

We use the term "consultation" to refer to choosing gene orders with high growth rates in each algorithm by considering attributes of classifiers from C5.0 [38] in order to avoid over-fitting. Such consultation algorithms are common in the field of artificial intelligence for games, such as Chess [39] and Shogi [40], in order to avoid over-fitting in each game engine and increase the overall performance. The problem becomes a classification domain, and the parameters of the classifier model were selected to maximize ten-fold cross-validation accuracy.

A. *Two-Algorithm Consultation*

Using LCS and C5.0, we classified 45 explanatory variables (describing the relative orders between two genes) into eight growth rate groups defined by an equal interval (Table I)—the growth rates take precise values, so they had to be converted into finite classes for the classification task. To test the classification performance in wet-lab experiments, we also examined 20 random datasets out of the total 144 datasets as a test dataset.

Based on these classification results, we selected six gene orders within the operon that are predicted to increase *E. coli* growth, and we then designed strains in order to test them using wet-lab experiments. First, we identified four gene orders that were classified as promoting growth rates exceeding 0.7/h in every algorithm considering C5.0 attributes. Furthermore, we selected two gene orders predicted to have growth rates that are relatively high but significantly different from those of the original dataset in terms of the arrangement

TABLE II
CLASSES OF GROWTH RATES FOR FOUR-ALGORITHM CONSULTATION

Classes	Growth Rate	Classes	Growth Rate
0	0–0.4/h	0.4	0.4–0.5/h
0.5	0.5–0.6/h	0.6	0.6–0.65/h
0.65	0.65–0.7/h	0.7	0.7–0.72/h
0.72	$\geq$ 0.72/h		

of genes to investigate the influence of modifying gene order remarkably.

B. *Four-Algorithm Consultation*

In addition to using LCS and C5.0, we also exploited random forest [41], which is an efficient ensemble learning method that employs many decision trees, and multilayer perceptron [42], which is a standard neural network model. We classified 45 explanatory variables (again, describing the relative orders between two genes) into seven growth rate groups (Table II). Each growth rate is divided into seven classes with smaller ranges for high growth rates in order to predict gene orders with high growth rates more accurately. The results of the previous experiment (six datasets) were also used as the main dataset for this analysis and 21 additional datasets from the total 150 datasets were used as the test dataset; three datasets per class were employed as test data.

From these classification results, we selected six gene orders for the operon structure and designed them for subsequent wet-lab experiments. First, we identified two gene orders that were estimated to promote growth rates in excess of 0.72/h by our LCS analysis and 0.7/h in the C5.0 analysis and random forest analysis, which considers the C5.0 attributes. In addition, we selected two gene orders that were estimated to promote growth rates in excess of 0.7/h by our LCS, C5.0, and random forest analyses and in excess of 0.65/h by our multilayer perceptron analysis, which considers C5.0 attributes. Finally, two gene orders were identified that were classified as promoting growth rates in excess of 0.7/h by our LCS and random forest analyses, 0.65/h by our C5.0 analysis, and 0.72/h by our multilayer perceptron analysis, which considers C5.0 attributes.

V. RESULTS

This section shows how our consultation algorithms utilizing LCS work in cases of consultation using two and four algorithms. The results include both computational simulations and their biological verification. Our method performed effective data mining in this gene expression analysis owing to the high accuracy of LCS and its ability to determine definite understandable rules that describe complex systems efficiently.

A. *Two-Algorithm Consultation*

Classification of Test Data by LCS. Fig. 5 shows classification results and the numbers of gene orders are presented with

(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	<-classified as
1				1				(a): Growth 0 ～
								(b): Growth 0.1 ～
				1				(c): Growth 0.2 ～
			1	5				(d): Growth 0.3 ～
			1	2	3	1		(e): Growth 0.4 ～
			1	1	1	1		(f): Growth 0.5 ～
								(g): Growth 0.6 ～
								(h): Growth 0.7 ～

Bold: classified correctly

Fig. 5. **Classification of test data** by learning classifier systems.

(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	<-classified as
1	1							(a): Growth 0 ～
								(b): Growth 0.1 ～
1								(c): Growth 0.2 ～
	1		1	4				(d): Growth 0.3 ～
	1		1	4		1		(e): Growth 0.4 ～
				2	2			(f): Growth 0.5 ～
								(g): Growth 0.6 ～
								(h): Growth 0.7 ～

Bold: classified correctly

Fig. 6. **Classification of test data** by C5.0.

both the classified class and the actual class. Classification succeeded with 25% accuracy; if up to one class error is allowed, we achieved 80% classification accuracy. Most gene orders were classified as promoting growth rates between 0.4/h and 0.5/h.

According to a rule set of eight simple and interpretable rules, a gene order was determined that was given the highest class assignment ($\geq$ 0.7/h; the $\rightarrow$ operator represents the rule stating that the gene preceding the operator is located in front of the gene following the operator; for example, A$\rightarrow$B means gene A is located before gene B). According to this rule set, gene A tends to be assigned to the front of the operon, while gene J tends to be assigned to the back. The rule set inferred to describe the highest growth rate classification using LCS is as follows.

- A$\rightarrow$B, A$\rightarrow$G, B$\rightarrow$H, C$\rightarrow$I, D$\rightarrow$F, E$\rightarrow$I, E$\rightarrow$J, H$\rightarrow$J

Classification of Test Data by C5.0. C5.0 produced classifications with 40% accuracy (Fig. 6); permitting one error class, this method obtained 80% of classification accuracy. Six gene orders promoting growth rates between 0.3/h and 0.4/h or between 0.5/h and 0.6/h were incorrectly classified as promoting growth rates between 0.4/h and 0.5/h.

Growth Rates of Newly Designed Operons. We selected six novel gene orders within the operon that were classified by our two-algorithm consultation to promote high growth rates and then we designed them for wet-lab verification experiments. They exhibited high growth rates (Fig. 7) and

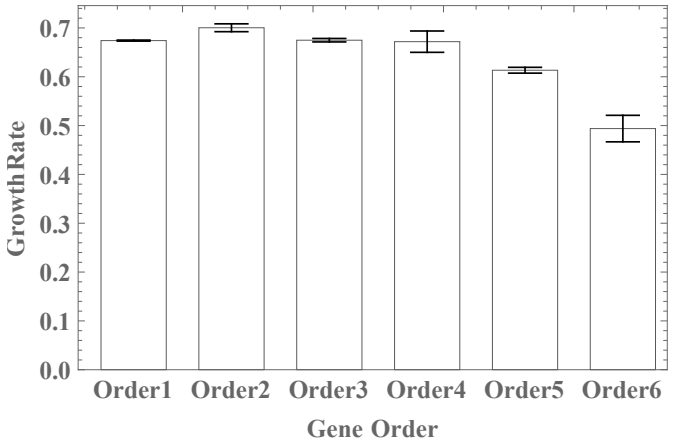

Fig. 7. **Growth rates of *E. coli*** with novel gene orders shown as means ± SD. Strains (Gene orders): Order1 (ABEICHDFGJ); Order2 (ABDCEGFIHJ); Order3 (ABCDEIHFGJ); Order4 (ABCDFGEIHJ); Order5 (AGBIEJCHDF); and Order6 (AEICHDFBGJ).

(a)	(b)	(c)	(d)	(e)	(f)	(g)	<-classified as	
	3						(a): Growth 0	~
1	1	1					(b): Growth 0.4	~
	3						(c): Growth 0.5	~
			1	2			(d): Growth 0.6	~
			1	2			(e): Growth 0.65	~
					3		(f): Growth 0.7	~
				1	1	1	(g): Growth 0.72	~

Bold: classified correctly

Fig. 8. **Classification of test data** by learning classifier systems.

the empirical error was less than 6% per sample. In particular, Order2 gave a growth rate comparable to the highest in the dataset (around 0.73/h) considering that the maximum standard deviation of our dataset is around 0.05/h. Order5 and Order6, which each have completely different gene orders from the dataset, demonstrated low growth rates compared with the other orders.

B. Four-Algorithm Consultation

Classification of test data by LCS. Fig. 8 shows classification results and the numbers of gene orders are presented with both the classified class and the actual class. LCS yielded classifications that succeeded with around 38% accuracy (Fig. 8); if up to one class error is allowed, we achieved around 95% classification accuracy. Six gene orders promoting growth rates between 0/h and 0.4/h or between 0.5/h and 0.6/h were classified incorrectly as promoting growth rates between 0.4/h and 0.5/h.

This analysis produced a restrictive rule set of 13 rules that determined a gene order that was assigned to the highest class ($\geq$0.72/h). Genes A, B, C, and D each tended to be assigned to the front of the operon, while gene J tended to be assigned to the back. The rank order of gene A has an especially strong

(a)	(b)	(c)	(d)	(e)	(f)	(g)	<-classified as	
1	2						(a): Growth 0	~
1	1	1					(b): Growth 0.4	~
		3					(c): Growth 0.5	~
				3			(d): Growth 0.6	~
				3			(e): Growth 0.65	~
					3		(f): Growth 0.7	~
				3			(g): Growth 0.72	~

Bold: classified correctly

Fig. 9. **Classification of test data** by C5.0.

(a)	(b)	(c)	(d)	(e)	(f)	(g)	<-classified as	
1	2						(a): Growth 0	~
1	1	1					(b): Growth 0.4	~
	1	2					(c): Growth 0.5	~
				3			(d): Growth 0.6	~
			1	2			(e): Growth 0.65	~
					3		(f): Growth 0.7	~
					2	1	(g): Growth 0.72	~

Bold: classified correctly

Fig. 10. **Classification of test data** by random forest.

influence on the growth rate. The rule set inferred to describe the highest growth rate classification using LCS is as follows.

- A→B, A→C, B→D, B→G, C→G, C→I, D→H, E→I, F→J, G→J, H→E, H→F, I→J

Classification of Test Data by C5.0. C5.0 produced a classification with approximately 52% accuracy (Fig. 9); within one class error, we obtained 86% classification accuracy. Six gene orders promoting growth rates between 0.6/h and 0.65/h or exceeding 0.72/h were incorrectly classified as promoting growth rates between 0.65/h and 0.7/h.

Classification of Test Data by Random Forest. The random forest analysis performed classification with around 48% accuracy (Fig. 10); if up to one class error is allowed, this method reached 100% classification accuracy. Three gene orders promoting growth rates between 0.6/h and 0.65/h were incorrectly classified as promoting growth rates between 0.65/h and 0.7/h.

Classification of Test Data by Multilayer Perceptron. The multilayer perceptron classification yielded 57% accuracy (Fig. 11); within one class error, this method achieved 90% classification accuracy. Five gene orders promoting growth rates between 0.6/h and 0.65/h or exceeding 0.7/h were incorrectly classified as promoting growth rates between 0.65/h and 0.7/h.

Growth Rates of Newly Designed Operons. We designed six novel operons with gene orders predicted to have high growth rates according to our four-algorithm consultation. As illustrated in Fig. 12, all of the newly designed operons showed high growth rates ($>$0.6/h) and the empirical error was less

(a)	(b)	(c)	(d)	(e)	(f)	(g)	<-classified as	
2	1						(a): Growth 0	～
1	2						(b): Growth 0.4	～
1		2					(c): Growth 0.5	～
			3				(d): Growth 0.6	～
				1	2		(e): Growth 0.65	～
				1	2		(f): Growth 0.7	～
				1		2	(g): Growth 0.72	～

Bold: classified correctly

Fig. 11. **Classification of test data** by multilayer perceptron.

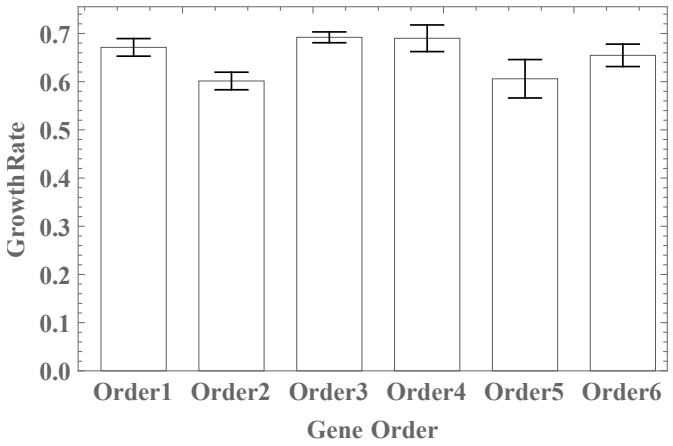

Fig. 12. **Growth rates of novel *E. coli* strains** presented as means ± SD. Strains (Gene orders): Order1 (ABCDFEHIGJ); Order2 (ACBDFGEHIJ); Order3 (ABDCEFIGHJ); Order4 (ABDCGEHFIJ); Order5 (ABCDFEIGHJ); and Order6 (ABCDFIEGHJ).

than 7% per sample. However, no operon promoted a higher growth rate than that which was obtained by the previous experiment based off of two-algorithm consultation. Order5, as shown in Fig. 12, demonstrated a large standard deviation (approximately 0.04). The gene orders obtained from the four-algorithm analysis are more similar to each other compared with those obtained in the two-algorithm analysis as it requires more restrictions on selection.

VI. Conclusion

We found that certain two-gene order rules can be used to design operons that significantly effect *E. coli* growth (i.e., production) using consultation algorithms that include LCS. Moreover, we also successfully created new *E. coli* strains with high growth rates using these operon construction rules. Genes that are closer to the promoter in an operon exhibit higher mRNA expression in general, as supported by real-time RT-PCR results. However, the explanation for severe growth rate differences among strains with different novel operons is unclear. The interactions between genes may differ substantially as their sequential order changes. Yet, other potential explanations of the relationship between gene order and growth rate warrant consideration.

Most operon rearrangement studies without computational predictive modeling suggest that the gene orders that resemble those of wild-type strains tend to have high growth rates [6], [7]. Our computationally optimized operons are similar to wild-type strains in gene order to some extent, as expected. Gene A, which exhibits much higher mRNA expression levels than other genes, was consistently located in the front of the operon in order to obtain a high growth rate. However, except for several genes strongly linked to their positions, reordering genes for growth optimization is possible. LCS rule sets, such as the optimal spatial relationship between gene E and gene I, provide further details on how gene orders can be optimized. While this is difficult to predict experimentally, the results are easily understood by biologists. However, if understandable rules are ignorerd, random forest might be the most suitable method as it achieves 100% accuracy if up to one-class incorrect classification is allowed.

Our study also reveals that surpassing the highest growth rate of the dataset is challenging for the following reasons: (1) although the classification of test data went well, no operon promoted growth rates exceeding those previously found; (2) the dataset is small and noisy; and (3) the dataset is biased and efficient operons share similar gene orders.

We aimed to optimize operon structure for the largest number of genes thus far—ten genes, which can be rearranged into 3,628,800 orders—by a novel computational approach. We focused on relative orders between two genes as the explanatory variables. However, more comprehensive variables, such as those used in natural language processing, may enhance classification accuracy; the structure involved in assessing sentences through word order is similar to the operon structure involved in predicting growth rates. These findings should also be confirmed using more randomized data to avoid over-fitting, especially when the number of genes within an operon is more than ten. Furthermore, we focused on operons that promote efficient growth; however, future studies should also explore operons that inhibit *E. coli* growth.

Taken together, our findings illustrate that machine learning—especially the use of consultation algorithms utilizing LCS to avoid over-fitting—can help identify the most efficiently structured operons even when the number of genes within an operon is large. Changes in mRNA expression of genes and gene interactions altered by gene order may cause these results. Computational results must be interpreted with caution, but newly designed operons tested in wet-lab experiments support this approach. This first computational study proves that pair-wise order relationships between genes produce significant differences in operon efficiency; given the difficulty of understanding all interactions between genes, future studies with more comprehensive explanatory variables are needed. Furthermore, our study suggests that LCS can play a significant role in data mining from large and noisy datasets extracted from biological systems, especially gene expression analysis for the mass-production of useful materials in synthetic biology.

REFERENCES

[1] E. Andrianantoandro, S. Basu, D. K. Karig, and R. Weiss, "Synthetic biology: New engineering rules for an emerging discipline," *Mol. Sys. Bio.*, vol. 2, 2006.

[2] W. Weber and M. Fussenegger, "Emerging biomedical applications of synthetic biology," *Nat. Rev. Genet.* vol. 13, pp. 21–35, 2012.

[3] Y. Y. Chen and C. D. Smolke, "From DNA to targeted therapeutics: Bringing synthetic biology to the clinic," *Sci. Transl. Med.* vol. 3, 2011.

[4] S. K. Lee, H. Chou, T. S. Ham, T. S. Lee, and J. D. Keasling, "Metabolic engineering of microorganisms for biofuels production: From bugs to synthetic biology to fuels," *Curr. Opin. Biotechnol.* vol. 19, pp. 556–563, 2008.

[5] K. Tsuge, K. Matsui, and M. Itaya, "One step assembly of multiple DNA fragments with a designed order and orientation in *Bacillus subtilis* plasmid," *Nucleic Acids Res.* vol. 31, 2003.

[6] Y. Nakagawa, K. Yugi, K. Tsuge, M. Itaya, H. Yanagawa, *et al.*, "Operon structure optimization by random self-assembly," *Nat. Comput.* vol. 9, pp. 173–181, 2010.

[7] T. Nishizaki, K. Tsuge, M. Itaya, N. Doi, and H. Yanagawa,"Metabolic engineering of carotenoid biosynthesis in *Escherichia coli* by ordered gene assembly in *Bacillus subtilis*," *Appl. Environ. Microbiol.* vol. 73, 2007.

[8] A. Hiroe, K. Tsuge, C. T. Nomura, M. Itaya, and T. Tsuge, "Rearrangement of gene order in the *phaCAB* operon leads to effective production of ultrahigh-molecular-weight poly[(R)-3-Hydroxybutyrate] in genetically engineered *Escherichia coli*," *Appl. Environ. Microbiol.* vol. 78, 2012.

[9] J. H. Holland, "*Adaptation in natural and artificial system: An introduction with application to biology, control and artificial intelligence*," University of Michigan Press, 1975.

[10] J. H. Holland, "Adaptive algorithms for discovering and using general patterns in growing knowledge bases," *Int. J. Pol. Anal. Inform. Syst.* vol. 4, pp. 245–268, 1980.

[11] J. H. Holland and J. S. Reitman, "Cognitive systems based on adaptive algorithms," in *Pattern Directed Inference Systems* Academic Press, 1978, pp. 313–329.

[12] P. M. Lee and T. C. Hsiao, "Applying LCS to affective image classification in spatial-frequency domain," in *Proc. IEEE C. Evol. Computat.* IEEE, 2014, pp. 1690–1697.

[13] J. Bacardit, M. Stout, J. D. Hirst, A. Valencia, R. E. Smith, *et al.*, "Automated alphabet reduction for protein datasets," *BMC Bioinformatics* vol. 10, 2009.

[14] F. Kharbat, M. Odeh, and L. Bull, "Knowledge discovery from medical data: An empirical study with XCS," in *Learning Classifier Systems in Data Mining* Springer, 2008, pp. 93—121.

[15] J. Bacardit, E. K. Burke, and N. Krasnogor, "Improving the scalability of rule-based evolutionary learning," *Memetic Computing* vol. 1, pp. 55–67, 2009.

[16] M. A. Franco, N. Krasnogor, and J. Bacardit, "Analysing BioHEL using challenging boolean functions," *Evol. Intell.* vol. 5, pp. 87–102, 2012.

[17] C. A. Rabinovitch-Deere, J. W. Oliver, G. M. Rodriguez, and S. Atsumi, "Synthetic biology and metabolic engineering approaches to produce biofuels," *Chem. Rev.* vol. 113, pp. 4611–4632, 2013.

[18] J. C. Anderson, E. J. Clarke, A. P. Arkin, and C. A. Voigt, "Environmentally controlled invasion of cancer cells by engineered bacteria," *J. Mol. Biol.* vol. 355, pp. 619–627, 2006.

[19] T. K. Lu and J. J. Collins, "Dispersing biofilms with engineered enzymatic bacteriophage," in *Proc. Natl. Acad. Sci. USA.* PNAS, 2007, pp. 11197–11202.

[20] H. B. Larman, Z. Zhao, U. Laserson, M. Z. Li, A. Ciccia, *et al.*, "Autoantigen discovery with a synthetic human peptidome," *Nat. Biotechnol.* vol. 29, pp. 535–541, 2011.

[21] M. Elowitz and W. A. Lim, "Build life to understand it," *Nature* vol. 468, pp. 889–890, 2010.

[22] A. A. Cheng and T. K. Lu, "Synthetic biology: An emerging engineering discipline," *Annu Rev Biomed Eng.* vol. 14, pp. 155–178, 2012.

[23] H. Al-Sahaf, M. Zhang, M. Johnston, and B. Verma Image Descriptor, "A genetic programming approach to multiclass texture classification," in *Proc. IEEE C. Evol. Computat.* IEEE, 2015, pp. 2460–2467.

[24] Y. Chen, Y. Shang, and D. Xu, "Multi-dimensional scaling and MODELLER-based evolutionary algorithms for protein model refinement," in *Proc. IEEE C. Evol. Computat.* IEEE, 2014, pp. 1038–1045.

[25] F. Ortuno, J. P. Florido, J. M. Urquiza, H. Pomares, A. Prieto, *et al.*, "Optimization of multiple sequence alignment methodologies using a multiobjective evolutionary algorithm based on NSGA-II," in *Proc. IEEE C. Evol. Computat.* IEEE, 2012, pp. 1–8.

[26] Y. Naruse, H. Hamada, T. Hanai, and H. Iba, "Evolutionary design of oscillatory genetic networks *in silico*," in *Proc. IEEE C. Evol. Computat.* IEEE, 2015, pp. 1596–1603.

[27] M. S. Nobile and H. Iba, "A double swarm methodology for parameter estimation in oscillating gene regulatory networks," in *Proc. IEEE C. Evol. Computat.* IEEE, 2015, pp. 2376–2483.

[28] E. Glaab, J. Bacardit, J. M. Garibaldi, and N. Krasnogor, "Using rule-based machine learning for candidate disease gene prioritization and sample classification of cancer gene expression data," *PLoS ONE* vol. 7, 2012.

[29] A. Zibakhsh and M. S. Abadeh, "Gene selection for cancer tumor detection using a novel memetic algorithm with a multi-view fitness function," *Eng. Appl. Artif. Intell.* vol. 26, pp. 1274–1281, 2013.

[30] M. Abedini, M. Kirley, and R. Chiong, "Incorporating feature ranking and evolutionary methods for the classification of high-dimensional DNA microarray gene expression data," *Australas. Med. J.* vol. 6, pp. 272–279, 2013.

[31] H. N. Lim, Y. Lee, and R. Hussein, "Fundamental relationship between operon organization and gene expression," in *Proc. Natl. Acad. Sci. USA.* PNAS, 2011, pp. 10626–10631.

[32] M. M. White, "Pretty subunits all in a row: using concatenated subunit constructs to force the expression of receptors with defined subunit stoichiometry and spatial arrangement," *Mol. Pharmacol.* vol. 69, pp. 407–410, 2006.

[33] M. Bujara, M. Schmperli, R. Pellaux, M. Heinemann, and S. Panke, "Optimization of a blueprint for *in vitro* glycolysis by metabolic real-time analysis," *Nat. Chem. Biol.* vol. 7, pp. 271–277, 2011.

[34] T. Rochat, P. Bouloc, and F. Repoila, "Gene expression control by selective RNA processing and stabilization in bacteria," *FEMS Microbiol. Lett.* vol. 344, pp. 104–113, 2013.

[35] C. Xu, R. Huang, L. Teng, X. Jing, J. Hu, *et al.*, "Cellulosome stoichiometry in Clostridium cellulolyticum is regulated by selective RNA processing and stabilization," *Nat. Comm.* vol. 6, 2015.

[36] B. F. Pfleger, D. J. Pitera, C. D. Smolke, and J. D. Keasling, "Combinatorial engineering of intergenic regions in operons tunes expression of multiple genes," *Nat. Biotechnol.* vol. 24, pp. 1027–1032, 2006.

[37] J. Bacardit, "*Pittsburgh genetics-based machine learning in the data mining era: representations, generalization, and run-time*," PhD thesis, Ramon Llull University, 2004.

[38] J. R. Quinlan, "*C4.5: Programs for machine learning*," Morgan Kaufmann, 1993.

[39] K. T. Spoerer, T. Okaneya, K. Ikeda, and H. Iida, "Further investigations of 3-member simple majority voting for Chess," in *Computers and Games* Springer, 2014, pp. 199—207.

[40] T. Obata, T. Sugiyama, K. Hoki, and T. Ito, "Consultation algorithm for computer Shogi: Move decisions by majority," in *Computers and Games* Springer, 2011, pp. 156—165.

[41] L. Breiman, "Random Forests," *Mach. Learn.* vol. 45, pp. 5–32, 2001.

[42] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.* vol. 2, pp. 359–366, 1989.