bioRxiv, 2022, pp. 1–9

doi: https://doi.org/10.1101/2022.10.15.512354
Advance Access Publication Date: Day Month Year
Problem Solving Protocol

PROBLEM SOLVING PROTOCOL

EVMP: Enhancing machine learning models for synthetic promoter strength prediction by Extended Vision Mutant Priority framework

Weiqin Yang ,[2,†] Dexin Li [2,†] and Ranran Huang [1,*]

[1] Institute of Marine Science and Technology, Shandong University, 72 Binhai Road, 266237, Qingdao, China and [2] School of Computer Science and Technology, Shandong University, 72 Binhai Road, 266237, Qingdao, China
*Corresponding author: Ranran Huang, Email: huangrr@sdu.edu.cn
[†] The first two authors contribute equally to the paper

Abstract

In metabolic engineering and synthetic biology applications, promoters with appropriate strengths are critical. However, it is time-consuming and laborious to annotate promoter strength by experiments. Nowadays many machine learning (ML) methods are applied to synthetic promoter strength prediction, but existing models are limited by the excessive proximity between synthetic promoters. In order to enhance ML models to better predict the synthetic promoter strength, we propose EVMP(Extended Vision Mutant Priority), a universal framework which utilize mutation information more effectively. In EVMP, synthetic promoters are equivalently transformed into base promoter and corresponding k-mer mutations, which are input into BaseEncoder and VarEncoder respectively. In Trc synthetic promoter library, EVMP was applied to multiple ML models and the model effect was enhanced to varying extents, up to 63.1%, while the SOTA(state-of-the-art) record was improved by 18.9%. EVMP also provides optional data augmentation based on multiple base promoters, which further improved the model performance by 22.2% compared with non-EVMP SOTA record. In further study, extended vision is shown to be essential for EVMP. We also found that EVMP can alleviate the over-smoothing phenomenon, which may contribute to its effectiveness. Our work suggests that EVMP can highlight the mutation information of synthetic promoters and significantly improve the prediction accuracy of strength. The source code is available at https://github.com/Tiny-Snow/EVMP.

Key words: EVMP, machine learning, promoter strength prediction, deep learning, synthetic promoter mutation library

1. Introduction

Promoters are the fundamental components of transcriptional regulation and have a direct impact on gene expression. In metabolic engineering and synthetic biology applications, promoters with desired strengths are critical [1]. However, it is challenging for them to meet the requirements of the logical design and optimization of metabolic pathway due to the insufficient number of well-characterized promoters. Therefore, a vast library with hundreds of promoters with better properties must be created and characterized [2].

The conventional methods for building promoter libraries rely on functional module combinations or random sequence mutation techniques. Methods based on random sequence mutations, such as error-prone PCR, are regarded as a straightforward and effective mutagenesis technique and have been successfully used to synthesize artificial promoters. Alper et al. [3], for instance, mutagenized the bacteriophage PL-promoter using error-prone PCR methods. Finally, 22 promoters with an intensity distribution range of 196 times were selected from nearly 200 promoter mutants. Zhao et al. [4] constructed and characterized a mutant library of Trc promoters (P_{trc}) using 83 rounds of mutation-construction-screening-characterization engineering cycles, and established a synthetic promoter library that consisted of 3665 different variants, displaying an intensity range of more than two orders of magnitude. Despite the availability of experimental methods, obtaining a small number of useful promoters from a random library typically necessitates time-consuming and laborious screening, and given the enormous number of possible sequence combinations, the effective identification of these useful promoters is largely constrained by a relatively small mutation library. For example, a 20-nucleotide sequence space represents $4^{20} = 1.1 \times 10^{12}$ possibilities, outnumbering even the largest bacterial libraries and most robust high-throughput screens. Therefore, obtaining the desired promoter through mutation, modification, or screening of existing promoters is challenging. In order to effectively guide how to search for new promoters in the huge potential sequence space, new methodologies should be developed to explore the relationship between promoter sequence and function more thoroughly [5].

© The Author 2022. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

Fortunately, prediction of biological problems has been shown to be amenable to machine learning (ML) techniques, as comprehensively reviewed by de Jongh et al. [6]. For instance, Chen et al. [7, 8] developed a ResNet-based neural network and greatly improved the predicting accuracy of drug-induced liver injury. Several recent studies have applied ML for promoter strength prediction. By varying a 50-nt region in the 5′ UTR of the HIS3 promoter, Cuperus et al. [9] generated 500000 variants for use as their training dataset for a convolutional neural network (CNN). This allowed them to predictably improve the expression of promoters with both random and native 5′ UTRs. They also demonstrated the advantage of using synthetic libraries as they found motifs that enhance transcription that are absent from the yeast genome. Additionally, ML has also enabled the prediction of transcriptional outputs for constitutive, inducible, or synthetic promoters in *Saccharomyces cerevisiae*. For instance, training a CNN with diversified libraries over 10^5 in size, created by precisely altering the P_{GPD} constitutive promoter and a P_{ZEV}-based inducible system, allowed prediction of promoter strengths with an accuracy of 79% [10, 11]. Similarly, de Boer et al. [12] created what is perhaps the largest synthetic library ever made for ML promoter engineering by synthesizing over 100 million random 80-bp upstream activating sequences (UASs). This library was used as a training dataset for a TF-motif-based model that was able to correctly predict the expression level of 94% of random promoters. In an elegant combination of hybrid promoter engineering and ML in human cells, Wu et al. [13] created 129-bp UASs that contained tandem, human TF binding sites that represented over 6000 unique motifs taken from two TF databases, and then used the resulting dataset to train a generalized linear model with elastic net regularization (GLMNET) [13–15]. The model enables us to successfully predict the differential expression of individual promoters across different cell lines, a difficult task for mammalian cell engineering.

Despite the substantial progress the aforementioned studies have made, they still have some limitations. The existing methods basically regard synthetic promoter strength prediction tasks as simple natural language processing (NLP) tasks, which have completely different patterns and characteristics. For example, the length of the promoter sequence is often longer than that of the natural language sequence, and the promoter sequence only has four bases in its vocabulary. Moreover, in NLP tasks, even in some such as retrieving-based question answering, mutation information is not taken into account. Previous work has studied the effect of a large number of ML models. For example, Zhao et al. [4] compared the prediction accuracy of LSTM, Random Forests, GBDT, XGBoost, etc. on the Trc synthetic promoter library, and believed that XGBoost had the best effect. However, most of these are conventional ML models, more deep learning models and more methods targeting synthetic promoters need to be explored.

In this paper, we propose a novel framework, EVMP (Extended Vision Mutant Priority), to extract important mutation features and enhance ML models. In EVMP, synthetic promoters are equivalently transformed into EVMP format data, including base promoter and k-mer mutations [16], which are input into BaseEncoder and VarEncoder respectively. In Trc synthetic promoter library constructed by Zhao et al. [4], EVMP models had different extents of improvement compared with non-EVMP models, among which LSTM was improved by 63.1%, Transformer was improved by 32.2%, and the other models had effect improvement of 1.1% to 7.6%. Compared with the existing SOTA non-EVMP method, EVMP obtained better results and improved SOTA records by 18.9%. EVMP also provides optional data augmentation, which is based on selecting multiple different base promoters and obtaining multiple copies of data in EVMP format. Data augmentation further enhanced the effect of EVMP, which was 22.2% higher than that of SOTA non-EVMP. In addition, we conducted ablation experiments to demonstrate that extended vision, or the k-mer in mutation representation, is crucial for EVMP. We also discussed the source of the effectiveness of EVMP and found that EVMP can effectively alleviate the over-smoothing phenomenon that often occurs in synthetic promoter datasets. Our work suggests that EVMP can effectively improve the effect of ML models in synthetic promoter strength prediction. As far as we know, this is the first research work that integrates mutation features to predict promoter strength.

2. Methods

2.1. EVMP Architecture

As shown in Figure 1.A, EVMP framework receives EVMP format data as input and predicts the promoter strength, which mainly consists of BaseEncoder and VarEncoder as the backbone network. The pipeline of applying EVMP to synthetic promoter strength prediction task includes two steps: generating the EVMP format data from synthetic promoter dataset, and then passing the data through EVMP framework to obtain the prediction results.

2.1.1. EVMP Data Processing

In the synthetic promoter dataset, we first choose one promoter as the *base promoter* $S = (x_1, \cdots, x_n)$, where each x is a base $\in \mathbb{R}^d$ and $d = 5$ (one-hot encoding of four bases A, T, C, G, and B for the blank). For any synthetic promoter $S' = (x'_1, \cdots, x'_n)$, multiple mutation sites are obtained by pairing S' with the base promoter S, and the k-mer subsequences centered at each mutation site in synthetic promoter S' are called *k-mer mutations* $M(S, S')$. Finally, we obtain the EVMP format data consisting of base promoter S and k-mer mutations $M(S, S')$, which is equivalent to the original synthetic promoter S', denoted as $\langle S, M(S, S')\rangle$.

2.1.2. Data Augmentation

Data augmentation is optional in the EVMP pipeline, which is mainly implemented by multiple base promoters. In the case of without data augmentation, the base promoter defaults to P_{trc}, and each synthetic promoter S' has a one-to-one correspondence with a data $\langle P_{\mathrm{trc}}, M(P_{\mathrm{trc}}, S')\rangle$ in EVMP format. After applying data augmentation, we can select multiple (e.g. 10 in this experiment) base promoters $P_1, P_2, \cdots$, then for the same synthetic promoter S', multiple data $\langle P_1, M(P_1, S')\rangle, \langle P_2, M(P_2, S')\rangle, \cdots$ can be obtained. Selecting a suitable base promoter is usually difficult. Data augmentation can give the model more choices and the chance to select a better base promoter.

2.1.3. EVMP Framework

Similar to common NLP models, BaseEncoder deals with the base promoter directly. BaseEncoder takes the base promoter S as input and outputs the base embedding e_{base}.

$$e_{\mathrm{base}} := \mathrm{BaseEncoder}(S) \tag{1}$$

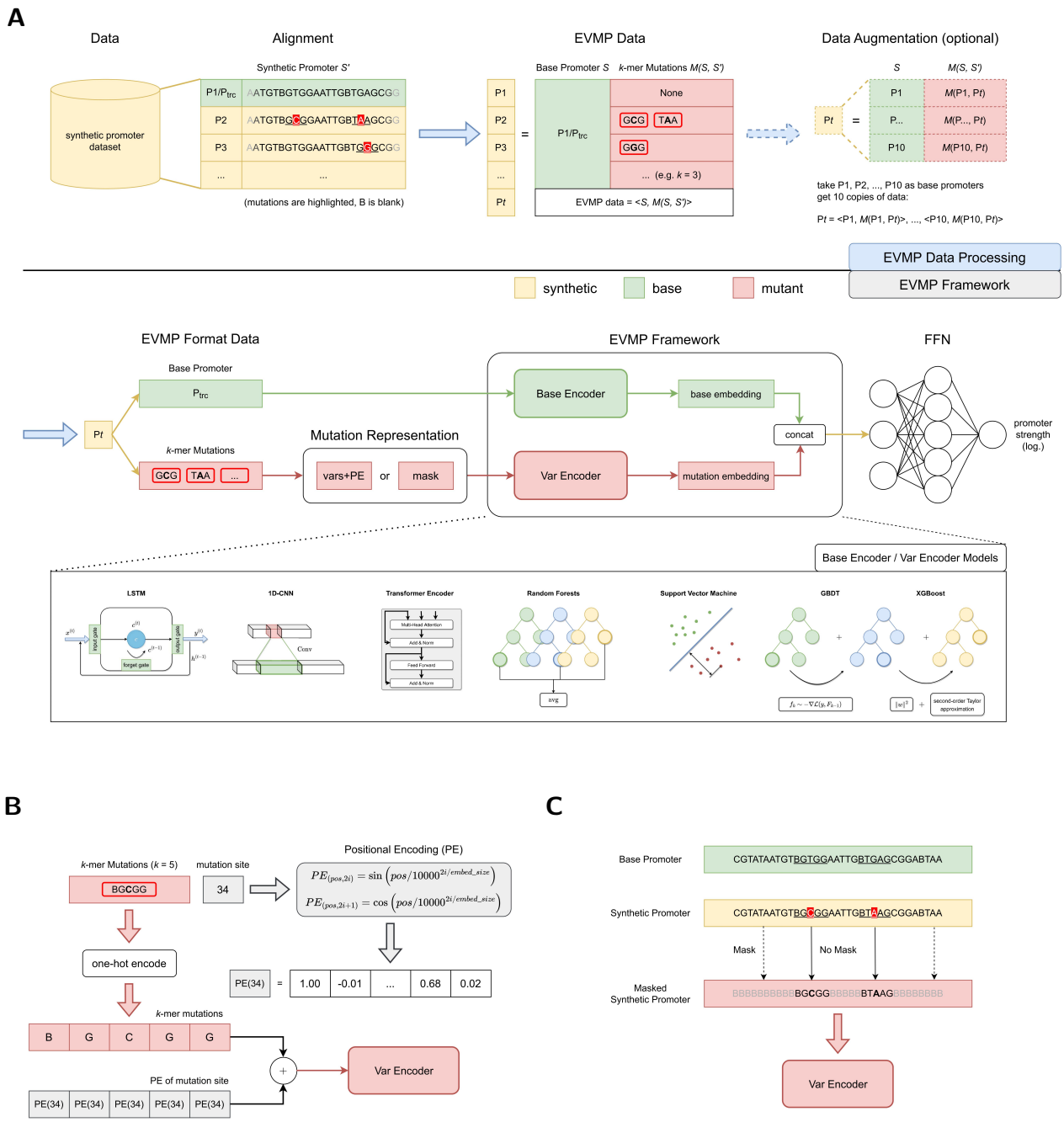

Fig. 1. Extended Vision Mutant Priority Framework. **(A)** The overall pipeline of EVMP framework for synthetic promoter strength prediction, which consists of data processing and framework. EVMP data processing is based on alignment, where the original synthetic promoter S' is equivalent to data in EVMP format: the base promoter S and the k-mer mutation $M(S, S')$. P_{trc} serves as the default base promoter, while data augmentation based on multiple base promoters is an optional step in data processing. EVMP framework consists of BaseEncoder and VarEncoder, which characterize the base promoter and the k-mer mutations after mutation representation, respectively. Base embedding and mutation embedding are then concatenated together and sent to a feed-forward network for strength prediction. **(B)** Vars+PE, one of the mutation representation paradigms. Each base in a k-mer mutation is added with the Positional Encoding of the mutation site. **(C)** Mask, one of the mutation representation paradigms. Each base that appears in a particular k-mer mutation is retained, and the rest is masked to B (blank).

Different from the common NLP models that deal with the whole synthetic promoter, VarEncoder deals with k-mer mutations $M(S, S')$ between base promoter S and synthetic promoter S'. VarEncoder receives k-mer mutations characterized by mutation representation module and outputs the mutation embedding e_{var}.

$$e_{\text{var}} := \text{VarEncoder}(M(S, S')) \tag{2}$$

In Section 2.2, we present two different mutation representation paradigms, including Vars+PE and Mask.

In the EVMP framework, BaseEncoder and VarEncoder are not specified to a particular model and are not even necessarily separate. In our work, alternative models for BaseEncoder and VarEncoder include LSTM [17, 18], Transformer [19], 1D-CNN [20, 21], Random Forests(RF) [22], GBDT [23], XGBoost [24] and SVM [25, 26]. For deep learning models, BaseEncoder and VarEncoder are separated, accepting inputs S and $M(S, S')$, respectively. For conventional machine learning models, BaseEncoder and VarEncoder are the same model, and the inputs S and $M(S, S')$ are concatenated together. In addition, the necessary mathematical foundations for the aforementioned machine learning methods are provided in Appendix 1. For more details on the model implementations, please refer to Appendix 2.

2.1.4. Promoter Strength Prediction

Base embedding e_{base} and mutation embedding e_{var} output by BaseEncoder and VarEncoder are concatenated and fed to a feed-forward network for promoter strength prediction. The model is trained on dataset $D = \{(x_i, y_i)|i = 1, \cdots, N\}$ with the objective function:

$$\min_{w} \sum_{i=1}^{N} |\hat{y}_i - y_i| + \lambda \|w\|^2 \tag{3}$$

where $\hat{y}_i$ is the output Log value of promoter strength of the EVMP format input x_i, w is the weights of model and λ is the regularization constant.

2.2. Mutation Representation Paradigms

Mutation representation is a critical component of EVMP. Different models may be suitable for different ways of representing mutations. But in general, mutation representation methods all need to satisfy the following two points: (i) The *extended vision* of mutation. (ii) The retention of mutation *positional information*. In the scope of our discussion, mutation representation methods are all based on k-mer mutations [16], that is, in line with extended vision in key point (i). Therefore, the key to achieving mutation representation lies in preserving positional information. In the following paragraphs, we introduce two different paradigms for mutation representation, including Vars+PE (used by LSTM and Transformer) and Mask (used by 1D-CNN and conventional ML models).

2.2.1. Vars+PE

In Vars+PE, the input of VarEncoder is the sequence of k-mer mutations, that is, multiple mutant subsequences. However, if this form is taken, the positional information of the individual k-mer mutations will be destroyed. One way to add position information for mutations is Positional Encoding (PE, Vaswani et al. [19]), a function $PE : \mathbb{R} \to \mathbb{R}^d$. Positional Encoding is generally added to the sequence directly, thus we call this technique *vars+PE*. In this work, we use sin-cos Positional Encoding in Definition 1.4 in Appendix 1.4 (Figure 1.B).

$$\begin{aligned} M_{\text{Vars+PE}}(S, S') = \{(&x'_{i-\lfloor k/2 \rfloor} + PE(i), \cdots, \\ &x'_{i+k-\lfloor k/2 \rfloor - 1} + PE(i))| \text{ if } x_i \neq x'_i, i = 1, \cdots, n\} \end{aligned} \tag{4}$$

It should be noted that BaseEncoder and VarEncoder add PE in different ways. BaseEncoder adds $PE(p)$ to the base at position p, bases at different positions are added different PE, while VarEncoder adds the same $PE(p')$ to k bases in a k-mer mutation at mutated position p', as shown in Equation (4).

2.2.2. Mask

In Section 2.2.1, PE is applied because the selection of k-mer mutations disrupted the initial order of bases. If the promoter is kept in its original form, there is no need to provide additional positional information. The *mask* technique fits this need, which assigns bases to B(Blank) if they never appear in any of the k-mer mutations so as to disguise the less significant bases.

$$\begin{aligned} M_{\text{Mask}}(S, S') = S' \cdot (\mathbb{I}(&x_{i-k+\lfloor k/2 \rfloor+1} \neq x'_{i-k+\lfloor k/2 \rfloor+1} \vee \\ &\cdots \vee x_{i+\lfloor k/2 \rfloor} \neq x'_{i+\lfloor k/2 \rfloor})|i = 1, \cdots, n) \end{aligned} \tag{5}$$

As shown in Figure 1.C, bases that appear in k-mer mutations are kept in their original positions, while the remaining bases are masked.

3. Experiments

The synthetic promoter strength prediction task is the primary experiment in this study. Our experiments are mainly based on the Trc synthetic promoter library created by Zhao et al. [4], which consisted of 3665 different synthetic promoters and used Log Fluorescence/OD_{600} to indicate promoter strength. Zhao et al. [4] experimented the results of LSTM, RF, XGBoost, GBDT and other models in the data. However, they did not set a separate test set, and they did not eliminate promoters with large differences (these promoters could not even be aligned). In our work, after sequence alignment with P_{trc} using MEGA [27], the synthetic promoters with large differences were discarded, and 3567 synthetic promoters were obtained, named $P_1, P_2, \cdots$. We randomly split the dataset into training, validation and test sets with an 8:1:1 ratio, all models have the same dataset partition. In addition, we added Transformer, 1D-CNN and SVM as choices of models. The results of EVMP and non-EVMP models on synthetic promoter strength prediction task are listed in Table 1. MAE (mean absolute error) and R^2 (coefficient of determination) are considered to be the evaluation metrics. Visual results are available at Figure A6 in Appendix 3.

3.1. EVMP Discards Redundant Bases

In the synthetic promoter library, most bases are identical between different promoters, and the models hardly gain information from the coincide bases. As shown in Figure 2.A, in comparison to P_{trc}, synthetic promoters with 2, 3 and 4 mutations account for the main part of the dataset, while the vast majority of synthetic promoters have no more than 8 mutations, which indicates that highlighting mutations is necessary.

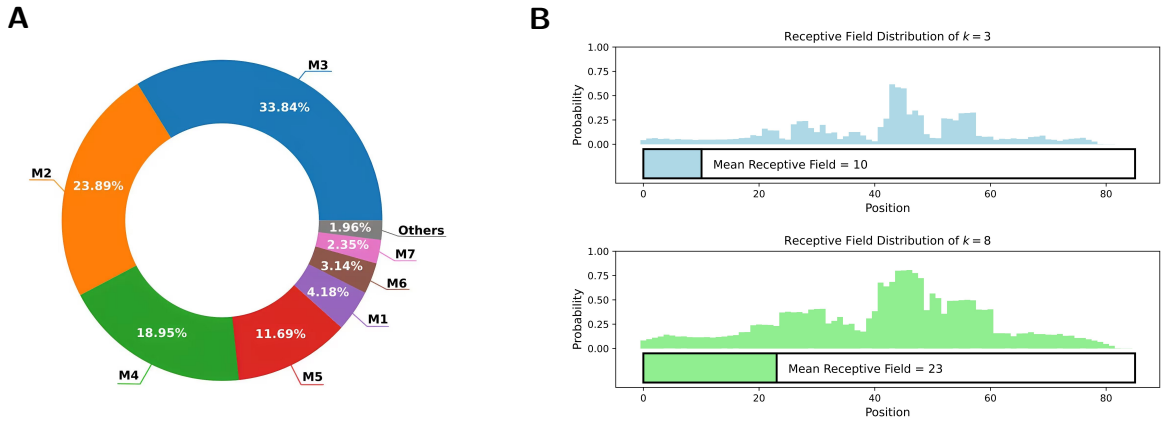

Fig. 2. (A) The proportion of different mutation times in 3567 synthetic promoters. Mt ($t = 1, 2, \cdots$) stands for synthetic promoters with t different mutation sites. The vast majority of synthetic promoters have no more than 8 mutation sites. **(B)** The probability that each position is included in the receptive field of VarEncoder, each promoter is filled B to 85 length. For $k = 3$ and 8, the mean receptive field of VarEncoder is 10 and 23, respectively, which is much less than the total length of the promoter.

Table 1. Results of EVMP and non-EVMP models on synthetic promoter strength prediction task. The entry *Mutant* is mutation representation method, k is the length of k-mer mutations, MAE is mean absolute error, R^2 is coefficient of determination, *Enhance* reflects the effect improvement of EVMP models compared to non-EVMP models, measured as the reduction in MAE.

Model	Mutant	k	Train		Valid			Test		
			MAE	R^2	MAE	R^2	Enhance↑	MAE	R^2	Enhance↑
LSTM	Vars+PE	3	0.505	-0.01	0.498	-0.01		0.497	0.00	
EVMP-LSTM			0.161	0.78	**0.202**	**0.73**	**+59.3%**	**0.184**	**0.79**	**+63.1%**
Transformer		5	0.281	0.55	0.290	0.45		0.273	0.61	
EVMP-Transformer			0.127	0.83	**0.181**	**0.76**	**+37.8%**	**0.185**	**0.77**	**+32.2%**
1DCNN(3 layers)	Mask	8	0.224	0.68	0.238	0.67		0.233	0.70	
EVMP-1DCNN(3 layers)			0.214	0.70	**0.227**	**0.70**	**+5.0%**	**0.230**	**0.71**	**+1.1%**
SVM			0.235	0.68	0.261	0.62		0.253	0.63	
GBDT			0.179	0.84	0.253	0.65		0.242	0.68	
XGBoost			0.145	0.89	0.242	0.66		0.231	0.69	
RF			0.084	0.95	0.227	0.63		0.226	0.64	
EVMP-SVM			0.210	0.72	**0.236**	**0.66**	**+9.3%**	**0.234**	**0.67**	**+7.6%**
EVMP-GBDT			0.157	0.87	**0.224**	**0.70**	**+11.3%**	**0.230**	**0.69**	**+4.8%**
EVMP-XGBoost			0.126	0.91	**0.218**	**0.70**	**+9.6%**	**0.226**	**0.69**	**+1.9%**
EVMP-RF			0.083	0.95	**0.223**	**0.68**	**+1.8%**	**0.222**	**0.68**	**+1.8%**

In EVMP, most of the same bases are useless and should be discarded. We define the *receptive field* of VarEncoder as the proportion of the number of bases appearing in a k-mer mutation to the length of promoter sequence. From Figure 2.B, when $k = 3$ and 8, the mean receptive fields of VarEncoder turn out to be 10 and 23 respectively, i.e. 11.8% and 27.1% of the whole promoter length. The input received by the VarEncoder discardes redundant bases while preserving the necessary information of mutations, which makes the effect of EVMP different from common NLP models.

3.2. EVMP Enhances the Effect of ML Models

The comparison of the effects of EVMP and non-EVMP models is listed in Table 1. From the results of test MAE, the SOTA (state-of-the-art) non-EVMP model was RF (0.226), and the effects of XGBoost (0.231) and 1D-CNN(0.233) were close to RF. Non-EVMP LSTM (0.497) and Transformer (0.273) were worse than other models. After applying EVMP, the effect of all models had different extents of improvement. Compared with non-EVMP models, the effect of LSTM (0.497 → 0.184) and Transformer (0.273 → 0.185) was improved by 63.1% and 32.2% respectively, which was far better than the effect of SOTA non-EVMP model (higher than 0.22), and it became a new SOTA record. In addition, other models also had a certain extent of effect improvement, ranging from 1.1% to 7.6%. The enhancement of EVMP makes the worst result of EVMP model (SVM, 0.234) even close to the SOTA result of non-EVMP (0.226). The experimental results show that EVMP can effectively improve the effect of each model on the synthetic promoter strength prediction task, and the effect of SOTA EVMP model (0.184) is much better than SOTA non-EVMP model (0.226), with a 18.9% increase.

3.3. Data Augmentation by Multiple Base Promoters Enhances EVMP

In the previous discussion, we did not apply optional data augmentation in EVMP. As described in Section 2.1.2, data augmentation is formed by selecting multiple base promoters

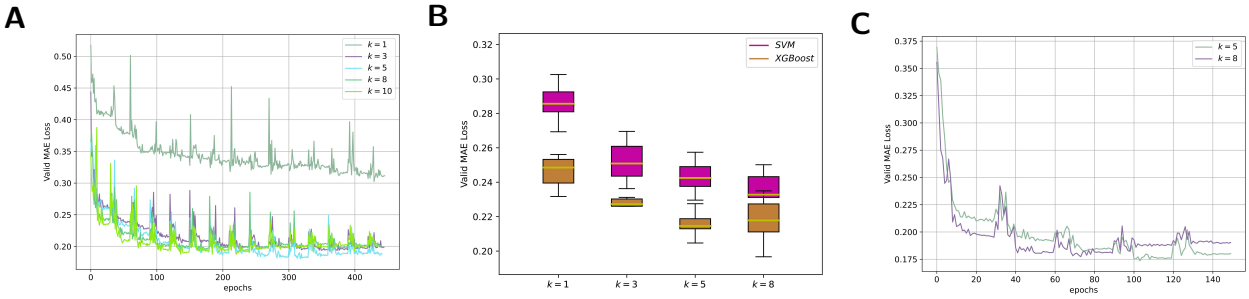

Fig. 3. Valid MAE of EVMP models with different k (length of k-mer Mutations). **(A)** Valid MAE curves for EVMP-Transformer with $k = 1, 3, 5, 8, 10$. As k increased, the effect of model first improved and then tended to be the same, $k = 5$ obtained the best result. **(B)** Valid MAE diagrams for EVMP-SVM and EVMP-XGBoost. As k increased, the model effect changed similarly to EVMP-Transformer. **(C)** Valid MAE curves for EVMP-Transformer after $\times$10-Rand data augmentation (see the definition in Section 3.3). The effect of $k = 5, 8$ was similar, but $k = 8$ converged faster.

$P_1, P_2, \cdots, P_{10}$ and obtaining multiple copies of data in EVMP format $\langle P_1, M(P_1, S')\rangle, \langle P_2, M(P_2, S')\rangle, \cdots, \langle P_{10}, M(P_{10}, S')\rangle$ for every synthetic promoter S'. Table A5 in Appendix 3 shows the 10 selected base promoters, which differ from P_{trc} by 2 to 7 different sites.

Three sets of controlled experiments related to data augmentation are presented in Table 2. Here, E1 and E3 are the previous non-EVMP and EVMP experiments without data augmentation, respectively. For convenience, we use P_{trc}, Rand, $\times$10 to denote the dataset using P_{trc} as base promoter, randomly using one of P_1 to P_{10} as base promoter, and using P_1 to P_{10} as base promoters (data augmentation), respectively. It should be noted that although the data with different base promoters are used, EVMP format data still represents the *same* synthetic promoters, which means that the datasets were exactly the same.

Table 2. Results of EVMP-Transformer in synthetic promoter strength prediction task with data augmentation. The entries *Train*, *Valid* and *Test* refer to the dataset processing method, where P_{trc} refers to using P_{trc} as base promoter, *Rand* refers to randomly selecting one of the 10 base promoters $P_1, \cdots, P_{10}$ as base promoter, and $\times$*10* refers to generating 10 copies of data by using all 10 base promoters. For visualization of the results, please refer to Figure A7 in Appendix 3. * E1 and E3 are the previous experiments as a control group.

ID	EVMP	Train	Valid & Test	Valid MAE	Test MAE
E1*		P_{trc}	P_{trc}	0.290	0.273
E2	✓	Rand	Rand	0.238	0.250
E3*	✓	P_{trc}	P_{trc}	0.181	0.185
E4	✓	$\times$10	P_{trc}	0.185	0.191
E5	✓	$\times$10	Rand	**0.174**	**0.176**

The first set of control experiments consisted of E1, E2 and E3. For each synthetic promoter of E2 (Rand-Rand), one of P_1 to P_{10} was randomly selected as base promoter. Note that different synthetic promoters can have different choices of base promoter. Results showed that the effect of Rand set (0.250) was not as good as that of P_{trc} set (0.185), which indicates that P_{trc} is an excellent enough base promoter, that is, the original template promoter of synthetic library should be considered as the first choice for base promoter. In addition, the effect of EVMP model using Rand set (0.250) was still better than non-EVMP model (0.273), which further illustrated the necessity of EVMP.

The second set of controlled experiments included E3 and E4. Compared to E3, E4 replaced the P_{trc} training set with $\times$10 training set for data augmentation. However, compared with the effect of P_{trc} set (0.185), the effect of the model was not improved (0.191), but almost the same. This indicates that there is no interference between data with different base promoters in EVMP data augmentation.

The third set of controlled experiments included E3, E4 and E5. Compared to E4 which used $\times$10 training set for data augmentation, E5 also used Rand validation & test sets. In the end, the effect after data augmentation (0.176) was better than the original effect (0.185), and a new SOTA record was formed, with a 22.2% increase compared to non-EVMP SOTA (0.226). This indicates that data augmentation is indeed effective, and model found more suitable among more choices of base promoters, thus further improving the effect of EVMP.

3.4. Extended Vision is Necessary for EVMP

We emphasized in Section 2.2 that extended vision is a necessary condition for EVMP, and k-mer mutations with $k > 1$ are an implementation of extended vision. However, we did not indicate why there should be $k > 1$, nor did we give a strategy for the determination of k value. It seems that the mutation input includes only those mutation sites (i.e. $k = 1$) is the least redundant approach. However, for promoter, one base may include too little information. As a result, a large number of biological software, such as Allesøe et al. [28] and Nurk et al. [29], have made use of k-mer. This is also confirmed by the experimental results. Figures 3.A and 3.B show the Valid MAE of EVMP-Transformer, EVMP-SVM and EVMP-XGBoost when different k is selected. The Valid MAE of EVMP-Transformer with $k = 5$ (0.181) was lower than that of $k = 1$ (0.302), 3 (0.193), 8 (0.193) and 10 (0.188). In particular, the effect of $k = 1$ was much worse than that of $k > 1$. EVMP-SVM and EVMP-XGBoost had similar conclusions, EVMP worked best when $k = 5$ and 8, and its effect was clearly different from that of EVMP with $k = 1$. The above experiments showed that extended vision is necessary for EVMP.

Different k may have little difference in effect, but its convergence rate is quite different. As shown in Figures 3.A and 3.C, a larger k (e.g. $k = 8$) will converge faster than a smaller k (e.g. $k = 5$) with little difference in effect. An empirical setting of k is that, if the average number of mutation sites is α, then $k = l/\alpha$, where l is the length of promoter. This will avoid too much overlap between k-mer mutations.

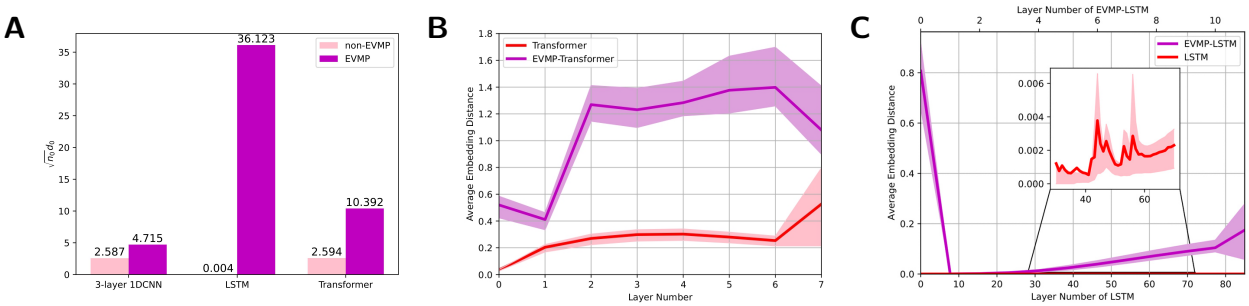

Fig. 4. Average embedding distance for each layer of deep learning models, with the 0-th layer serving as the input. Since BaseEncoder receives the same base promoter P_{trc}, non-EVMP models calculate the average embedding distance of each layer directly, while EVMP models calculate the average embedding distance of VarEncoder. **(A)** non-norm d_0 ($\sqrt{n_0}d_0$) for 1D-CNN(3 layers), LSTM and Transformer, which reflects the degree of dispersion of the input data. **(B)** Average embedding distance for each layer of Transformer and EVMP-Transformer. **(C)** Average embedding distance for each layer of LSTM and EVMP-LSTM, where LSTM calculates the average embedding distance for each time step.

4. Discussion

4.1. Why EVMP: Experimental Perspective

Experimental results have shown that EVMP can significantly enhance the effect of ML models in predicting the strength of synthetic promoters. However, why EVMP is effective remains to be further explored. We found that EVMP reduces the *over-smoothing* phenomenon, which may contributes to its effectiveness. Over-smoothing phenomenon occurs when the outputs of different inputs appear to be almost the same. Appropriate smoothing is regarded as a symbol of robustness in other tasks like image recognition [30]. However, considering the fact that the differences between synthetic promoters are much shallower than that between natural language sentences, over-smoothing phenomenon is likely to lead to a poor performance.

We define the *average embedding distance* to study over-smoothing phenomenon. Assume that synthetic promoters are $P_1, \cdots, P_N$, and P_0 is the base promoter (P_{trc}). The output embedding vector $e_l(P) \in \mathbb{R}^{n_l}$ are calculated for each layer l and each input P. Denote the average embedding distance d_l of each layer as follows:

$$d_l = \frac{1}{N}\sum_{i=1}^{N}\frac{\|(e_l(P_i) - e_l(P_0)\|_2}{\sqrt{n_l}} \tag{6}$$

Average embedding distance measures the average distance of each dimension of the embedding at the l-th layer, thus it is capable to represent the dispersion of the output of each layer in neural network.

In Figure 4, the average embedding distance of each layer was calculated for EVMP and non-EVMP implementations of 1D-CNN(3 layers), LSTM and Transformer, respectively, where LSTM calculated the average embedding distance of each time step. Figure 4.B showed that the average embedding distance of EVMP-Transformer was always much higher than that of non-EVMP Transformer. Figure 4.C showed that the growth rate of the average embedding distance of EVMP-LSTM from 0-th time step was much higher than that of non-EVMP LSTM. The above experimental results demonstrate that EVMP can effectively alleviate the over-smoothing phenomenon.

4.2. Why EVMP: Theoretical Perspective

Section 4.1 provides a comparison of the average embedding distance between neural network layers, in order to intuitively verify whether EVMP addresses the problem of over-smoothing representation and to elaborate the effectiveness of EVMP. Here, a theoretical illustration is provided to the rationality of this perspective.

Zou et al. [31] proved that when the width of each layer in deep ReLU networks is at least $\widetilde{\Omega}\left(n^{14}L^{16}/\phi^4\right)$, gradient descent can achieve zero training error within $\widetilde{O}\left(n^5L^3/\phi\right)$ iterations, where n is the number of training examples, L is the number of hidden layers, and ϕ is the maximum lower bound of the L_2-norm between any two training samples.

Here, we regard $\sqrt{n_0}d_0$ as an approximation of ϕ, where n_0 is the dimension of input. Figure 4.A illustrated ϕ of the three ML models, including 1D-CNN(3 layers), LSTM and Transformer, increased significantly after applying EVMP. Since the size of the training set n and the number of model layers L are remained unchanged, the overparameterizations of layer width and training iterations are reduced, and the model fitting effect tends to be improved. This statement is also enabled to be applied to RNN and CNN [31]. The above discussion shows that EVMP can indeed alleviate the over-smoothing phenomenon, which is indeed one reason for the effectiveness of EVMP.

4.3. EVMP Reduces the Need for Annotated Data under the Same Prediction Accuracy

Enhancing performance of ML models is just one aspect of EVMP. Another contribution of EVMP is to reduce the requirement for costly strength-annotated synthetic promoters under the original prediction accuracy. Here, given a fixed validation set proportion (1/10 of the original dataset), we randomly selected 1/9, 2/9, $\cdots$, 9/9 of the remaining 9/10 dataset as training set proportion, respectively. Figure 5 shows MAE and R^2 of training and validation sets for each EVMP-Transformer model.

According to our experimental results in Table 1, the SOTA MAE of non-EVMP model was no less than 0.22, while Figure 5 shows that EVMP only needed a training set of 5/9 original size to achieve this effect. In other words, EVMP only needed 56% of the original data while maintaining the same prediction accuracy, which greatly reduces the dependence on high-quality labeled data.

5. Conclusion

In this work, we proposed EVMP as a universal framework to enhance machine learning models for synthetic promoter

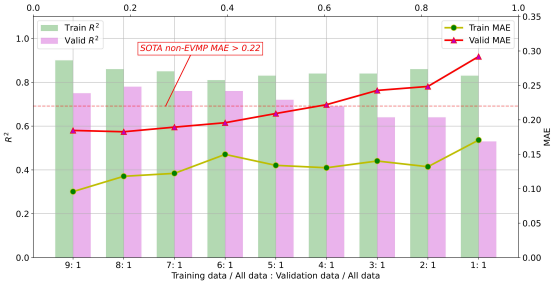

Fig. 5. As the proportion of the training set decreased, the MAE of the validation set increased and the R^2 decreased. When the proportion of training set to validation set is higher than 5:1, EVMP-Transformer achieved lower MAE than SOTA non-EVMP models.

strength prediction, which includes two steps: EVMP data processing and EVMP framework. The original synthetic promoter data are transformed into base promoter and k-mer mutations, which are processed by BaseEncoder and VarEncoder respectively to highlight mutation information. EVMP enhances many models and achieves new SOTA records. EVMP also provides optional data augmentation based on multiple base promoters, which can further improve the performance of EVMP. In further study, We experimentally verify that extended vision, or k-mer, is critical for the effectiveness of EVMP. In addition, EVMP is proved to be able to alleviate the over-smoothing phenomenon and thus improves the effect of models. In future research, other ways of mutation representation and information transfer between BaseEncoder and VarEncoder can be further explored.

Key Points

- We present EVMP, a universal framework for enhancing ML models on synthetic promoter strength prediction task. This method highlights mutation information of synthetic promoter and overcomes the bottleneck of the common NLP model method.
- EVMP can effectively improve the effect of each model to the level of SOTA non-EVMP model, and obtained a new SOTA record.
- EVMP provides a unique and convenient way of data augmentation, which can further enhance the effect of EVMP.
- EVMP reduces the over-smoothing phenomenon, which may be the reason why EVMP is effective.

6. Supplementary Material

Appendix is available online at https://www.biorxiv.org.

7. Funding

This work was supported by Grants 31970113 and 32170065 from the National Natural Science Foundation of China.

References

1. J. Gao, L. Jiang, J. Lian, Development of synthetic biology tools to engineer pichia pastoris as a chassis for the production of natural products, Synthetic and systems biotechnology 6 (2021) 110–119.
2. H. Tang, Y. Wu, J. Deng, N. Chen, Z. Zheng, Y. Wei, X. Luo, J. D. Keasling, Promoter architecture and promoter engineering in saccharomyces cerevisiae, Metabolites 10 (2020) 320.
3. H. Alper, C. Fischer, E. Nevoigt, G. Stephanopoulos, Tuning genetic control through promoter engineering, Proceedings of the National Academy of Sciences 102 (2005) 12678–12683.
4. M. Zhao, Z. Yuan, L. Wu, S. Zhou, Y. Deng, Precise prediction of promoter strength based on a de novo synthetic promoter library coupled with machine learning, ACS Synthetic Biology 11 (2021) 92–102.
5. A. P. Cazier, J. Blazeck, Advances in promoter engineering: novel applications and predefined transcriptional control, Biotechnology Journal 16 (2021) 2100239.
6. R. P. de Jongh, A. D. van Dijk, M. K. Julsing, P. J. Schaap, D. de Ridder, Designing eukaryotic gene expression regulation using machine learning, Trends in biotechnology 38 (2020) 191–201.
7. Z. Chen, Y. Jiang, X. Zhang, R. Zheng, R. Qiu, Y. Sun, C. Zhao, H. Shang, Resnet18dnn: prediction approach of drug-induced liver injury by deep neural network with resnet18, Briefings in Bioinformatics 23 (2022) bbab503.
8. Z. Chen, Y. Jiang, X. Zhang, R. Zheng, R. Qiu, Y. Sun, C. Zhao, H. Shang, The prediction approach of drug-induced liver injury: response to the issues of reproducible science of artificial intelligence in real-world applications, Briefings in Bioinformatics (2022).
9. J. T. Cuperus, B. Groves, A. Kuchina, A. B. Rosenberg, N. Jojic, S. Fields, G. Seelig, Deep learning of the regulatory grammar of yeast 5' untranslated regions from 500,000 random sequences, Genome research 27 (2017) 2015–2024.
10. R. S. McIsaac, P. A. Gibney, S. S. Chandran, K. R. Benjamin, D. Botstein, Synthetic biology tools for programming gene expression without nutritional perturbations in saccharomyces cerevisiae, Nucleic acids research 42 (2014) e48–e48.
11. B. J. Kotopka, C. D. Smolke, Model-driven generation of artificial yeast promoters, Nature communications 11 (2020) 1–13.
12. C. G. de Boer, E. D. Vaishnav, R. Sadeh, E. L. Abeyta, N. Friedman, A. Regev, Deciphering eukaryotic gene-regulatory logic with 100 million random promoters, Nature biotechnology 38 (2020) 56–65.
13. M.-R. Wu, L. Nissim, D. Stupp, E. Pery, A. Binder-Nissim, K. Weisinger, C. Enghuus, S. R. Palacios, M. Humphrey, Z. Zhang, et al., A high-throughput screening and computation platform for identifying synthetic promoters with enhanced cell-state specificity (specs), Nature communications 10 (2019) 1–10.
14. P. Kheradpour, M. Kellis, Systematic discovery and characterization of regulatory motifs in encode tf binding experiments, Nucleic acids research 42 (2014) 2976–2987.
15. M. T. Weirauch, A. Yang, M. Albu, A. G. Cote, A. Montenegro-Montero, P. Drewe, H. S. Najafabadi, S. A. Lambert, I. Mann, K. Cook, et al., Determination and inference of eukaryotic transcription factor sequence

specificity, Cell 158 (2014) 1431–1443.
16. W.-l. Liu, Q.-b. Wu, Analysis method and algorithm design of biological sequence problem based on generalized k-mer vector, Applied Mathematics-A Journal of Chinese Universities 36 (2021) 114–127.
17. S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.
18. F. A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: Continual prediction with lstm, Neural computation 12 (2000) 2451–2471.
19. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).
20. Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, L. Jackel, Handwritten digit recognition with a back-propagation network, Advances in neural information processing systems 2 (1989).
21. S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, D. J. Inman, 1d convolutional neural networks and applications: A survey, Mechanical systems and signal processing 151 (2021) 107398.
22. L. Breiman, Random forests, Machine learning 45 (2001) 5–32.
23. J. H. Friedman, Greedy function approximation: a gradient boosting machine, Annals of statistics (2001) 1189–1232.
24. T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.
25. C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (1995) 273–297.
26. B. E. Boser, I. M. Guyon, V. N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the fifth annual workshop on Computational learning theory, 1992, pp. 144–152.
27. S. Kumar, M. Nei, J. Dudley, K. Tamura, MEGA: A biologist-centric software for evolutionary analysis of DNA and protein sequences, Briefings in Bioinformatics 9 (2008) 299–306.
28. R. L. Allesøe, C. K. Lemvigh, M. V. Phan, P. T. Clausen, A. F. Florensa, M. P. Koopmans, O. Lund, M. Cotten, Automated download and clean-up of family-specific databases for kmer-based virus identification, Bioinformatics 37 (2021) 705–710.
29. S. Nurk, D. Meleshko, A. Korobeynikov, P. A. Pevzner, metaspades: a new versatile metagenomic assembler, Genome research 27 (2017) 824–834.
30. A. Ruderman, N. C. Rabinowitz, A. S. Morcos, D. Zoran, Pooling is neither necessary nor sufficient for appropriate deformation stability in cnns, arXiv preprint arXiv:1804.04438 (2018).
31. D. Zou, Y. Cao, D. Zhou, Q. Gu, Gradient descent optimizes over-parameterized deep relu networks, Machine learning 109 (2020) 467–492.

Wei-Qin Yang. is an undergraduate student at the School of Computer Science and Technology, Shandong University. His research interests include machine learning and bioinformatics.

De-Xin Li. is an undergraduate student at the School of Computer Science and Technology, Shandong University. His research interests include natural language processing and bioinformatics.

Ran-Ran Huang. is an associate professor at the Institute of Marine Science and Technology, Shandong University. His research interests include bioinformatics and Bacterial posttranscriptional regulation