bioRxiv, 2023, pp. 1–9

doi: https://doi.org/10.1101/2022.10.15.512354
Advance Access Publication Date: Day Month Year
Original Paper

ORIGINAL PAPER

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

Motivation: 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.
Results: 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 61.30%, while the SOTA(state-of-the-art) record was improved by 15.25%. EVMP also provides optional data augmentation based on multiple base promoters, which further improved the model performance by 17.95% 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 contributes to its effectiveness. Our work suggests that EVMP can highlight the mutation information of synthetic promoters and significantly improve the prediction accuracy of strength.
Availability and implementation: The source code is publicly available on github: https://github.com/Tiny-Snow/EVMP.
Contact: huangrr@sdu.edu.cn
Supplementary information: Appendix is available at *bioRxiv* online.

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 (Gao et al., 2021). 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 (Tang et al., 2020).

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. (2005), 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. (2021) 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 Author 2023. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

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 (Cazier and Blazeck, 2021).

Fortunately, prediction of biological problems has been shown to be amenable to machine learning (ML) techniques, as comprehensively reviewed by de Jongh et al. (2020). 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. (2017) 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% (McIsaac et al., 2014; Kotopka and Smolke, 2020). Similarly, de Boer et al. (2020) 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. (2019) 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) (Kheradpour and Kellis, 2014; Weirauch et al., 2014; Wu et al., 2019). The model enables us to successfully predict the differential expression of individual promoters across different cell lines, a difficult task for mammalian cell engineering.

Although the progress made in the aforementioned works is exciting, their methods are not without flaws. The key characteristic of the synthetic promoter dataset is that the synthetic promoters are too close to each other while their strengths exhibit a substantial degree of variation (Zhao et al., 2021). This results in the inability to differentiate the strengths of synthetic promoters through sequence homology, and commonly used language models such as LSTM also perform poorly. Therefore, we have considered another simple approach, which is to highlight the mutation information.

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 (Liu and Wu, 2021), which are input into BaseEncoder and VarEncoder respectively. We evaluated the effectiveness of EVMP on a Trc synthetic promoter library constructed by Zhao et al. (2021) and found that EVMP models exhibited varying degrees of improvement over non-EVMP models. Specifically, LSTM was improved by 61.30%, Transformer by 34.17%, and the other models had effect improvements ranging from 5.15% to 7.27%. Compared to the state-of-the-art (SOTA) non-EVMP method (Zhao et al., 2021), which was included in our experiments, EVMP achieved superior results and improved the SOTA non-EVMP record by 15.25%. EVMP offers optional data augmentation, which involves selecting multiple base promoters and generating multiple copies of data in the EVMP format. Data augmentation further enhanced the effectiveness of EVMP, which was 17.95% higher than that of the SOTA non-EVMP method. We conducted ablation experiments to demonstrate the critical role of the extended vision, or the k-mer in mutation representation, in EVMP framework. Additionally, we discussed the effectiveness of EVMP and found that it can alleviate the over-smoothing phenomenon that often occurs in synthetic promoter datasets. Our work suggests that EVMP can significantly improve the performance of ML models in predicting synthetic promoter strength, while its effectiveness is accompanied by reasonable interpretability. To the best of our knowledge, this is the first research to integrate mutation features for predicting promoter strength.

2. Methods

2.1. EVMP Architecture

As illustrated in Figure 1.A, EVMP framework takes EVMP format data as input and predicts promoter strength using BaseEncoder and VarEncoder as backbone networks. The pipeline for applying the EVMP framework to the synthetic promoter strength prediction task consists of two steps: generating EVMP format data from the synthetic promoter dataset and passing the data through the EVMP framework to obtain prediction results.

2.1.1. EVMP Data Processing

In the synthetic promoter dataset, a *base promoter* $S = (x_1, \cdots, x_n)$ is first selected, where each x is a base $\in \mathbb{R}^d$ and $d = 5$ (representing 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 referred to as *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 an optional step in the EVMP pipeline, which is primarily achieved through selecting multiple base promoters. When data augmentation is not applied, the base promoter is set to P_{trc} by default, and each synthetic promoter S' is mapped to a corresponding data point $\langle P_{\mathrm{trc}}, M(P_{\mathrm{trc}}, S')\rangle$ in the EVMP format. By introducing data augmentation, we can select multiple (e.g., 10 in this experiment) base promoters, denoted as $P_1, P_2, \cdots$, and generate multiple data points for the same synthetic promoter S'. Specifically, we can obtain $\langle P_1, M(P_1, S')\rangle, \langle P_2, M(P_2, S')\rangle, \cdots$, each corresponding to a different base promoter but the same synthetic promoter. Choosing a suitable base promoter is often challenging. Data augmentation provides the model with more choices and opportunities to select a better base promoter.

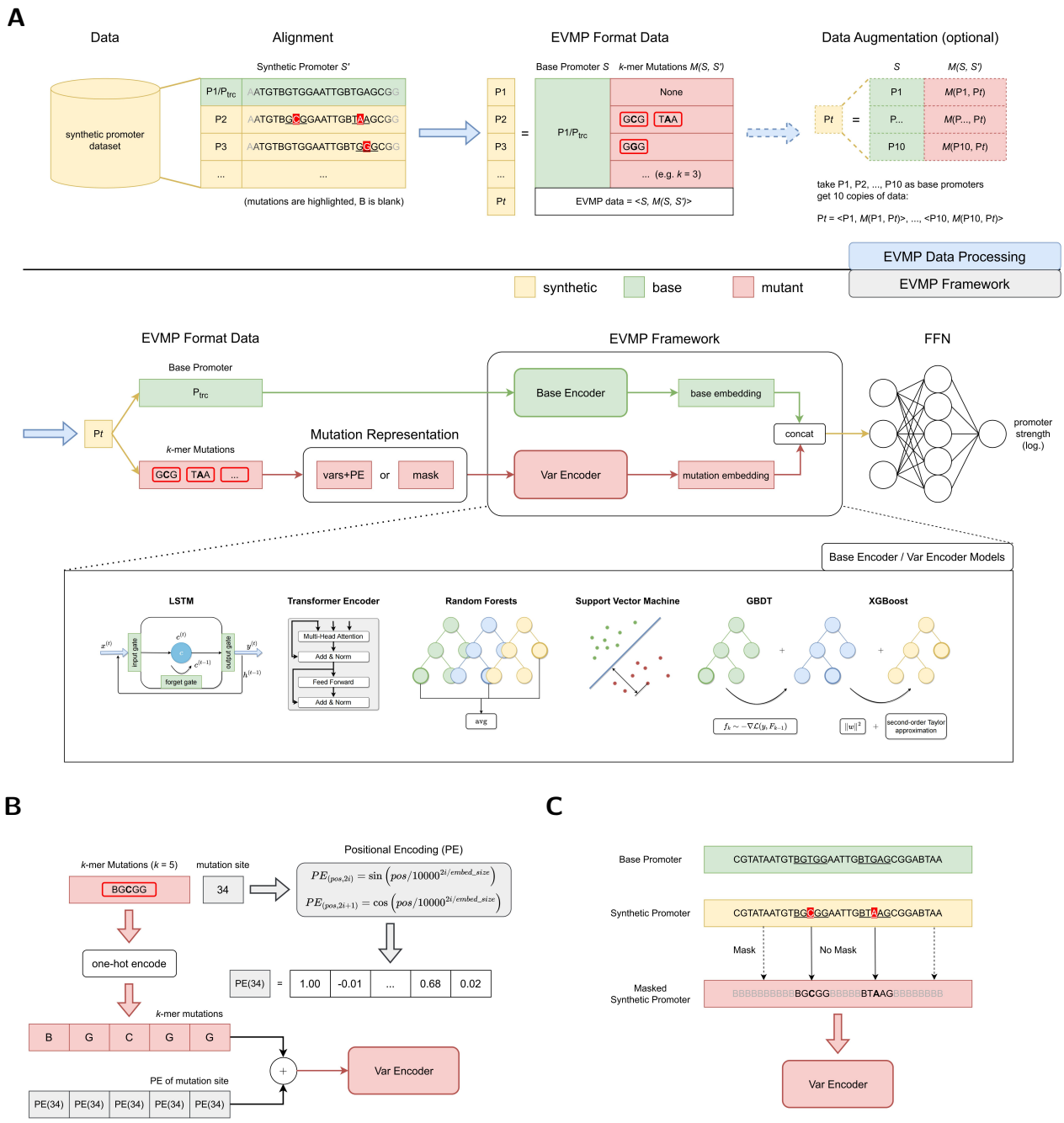

Fig. 1. Extended Vision Mutant Priority Framework. **(A)** The overall pipeline of EVMP framework for synthetic promoter strength prediction, which consists of data processing and framework. EVMP data processing is based on alignment, where the original synthetic promoter S' is equivalent to data in 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).

2.1.3. EVMP Framework

Similar to common natural language processing (NLP) models, the BaseEncoder in the EVMP pipeline directly processes the base promoter. Specifically, the BaseEncoder takes the base promoter S as input and produces the corresponding base embedding e_{base} as the output.

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

In contrast to typical NLP models that handle the entire synthetic promoter, the VarEncoder in the EVMP pipeline deals with k-mer mutations $M(S, S')$ between the base promoter S and the synthetic promoter S'. Specifically, the VarEncoder receives the k-mer mutations that are represented by the mutation representation module and produces the corresponding mutation embedding e_{var} as the output.

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

In Section 2.2, we will introduce two different mutation representation paradigms, namely Vars+PE and Mask.

In the EVMP framework, BaseEncoder and VarEncoder are model-agnostic and may not necessarily be distinct. Our study offers several alternatives for BaseEncoder and VarEncoder, including LSTM (Hochreiter and Schmidhuber, 1997; Gers et al., 2000), Transformer (Vaswani et al., 2017), Random Forests(RF) (Breiman, 2001), GBDT (Friedman, 2001), XGBoost (Chen and Guestrin, 2016), and SVM (Cortes and Vapnik, 1995; Boser et al., 1992). For deep learning models, BaseEncoder and VarEncoder can be separate, taking inputs S and $M(S, S')$, respectively. For conventional machine learning models, BaseEncoder and VarEncoder must be the same model, and the inputs S and $M(S, S')$ are concatenated. The implementation of the non-EVMP model is the BaseEncoder for each model. In addition, Appendix 1 provides the necessary mathematical foundations for the aforementioned machine learning methods. For more details on model implementations, see Appendix 2.

2.1.4. Promoter Strength Prediction

The base embedding e_{base} and mutation embedding e_{var} produced by the BaseEncoder and VarEncoder, respectively, are concatenated and passed through a feed-forward network (FFN) for predicting promoter strength. The model is trained on a dataset $D = \{(x_i, y_i)|i = 1, \cdots, N\}$ using 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 prediction 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 mutation representation methods may be appropriate for different models. Nevertheless, in general, all mutation representation methods must satisfy two fundamental requirements: (i) the *extended vision* of mutation, and (ii) the preservation of mutation *positional information*. In the scope of our discussion, all mutation representation methods are based on k-mer mutations, which aligns with the extended vision principle stated in requirement (i). Therefore, the key to achieving mutation representation lies in preserving positional information. In the following, we present two different paradigms for mutation representation, including Vars+PE (used by LSTM and Transformer) and Mask (used by conventional ML models).

2.2.1. Vars+PE

In Vars+PE, VarEncoder takes the sequence of k-mer mutations as input. However, using this form destroys the positional information of individual mutation sites. To incorporate positional information for mutations, one approach is to use Positional Encoding (PE, Vaswani et al. (2017)), which is a function $PE : \mathbb{R} \to \mathbb{R}^d$. Generally, the Positional Encoding is directly added to the sequence, and therefore, we refer to this technique as *vars+PE*. In this study, we utilize the sin-cos Positional Encoding defined in Definition 1.4 in Appendix 1.4 and 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. Specifically, 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, we utilized the PE technique since the selection of k-mer mutations resulted in the disruption of the initial base order. However, if the promoter sequence is maintained in its original form, the provision of additional positional information is unnecessary. 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

3.1. Datasets and Models

The central experiment of this study is the prediction of synthetic promoter strength. The experiments are primarily based on the Trc synthetic promoter library created by Zhao et al. (2021), which used log Fluorescence/OD600 as a measure of promoter strength. Previous research, such as that conducted by Zhao et al. (2021), has evaluated the performance of various models on this dataset, including LSTM, RF, XGBoost and GBDT. Our non-EVMP baselines had included all these competitive models in previous work. In addition, models such as Transformer and SVM were added as options. For EVMP models, BaseEncoder and VarEncoder were the same, and their implementation referred to Section 2.1.3. EVMP is based on alignment, and we used MEGA (Kumar et al., 2008) to perform alignment between the base promoter and synthetic promoter.

Our main experiment used a 9:1 split of the dataset as training data and test data, and the training data was split into

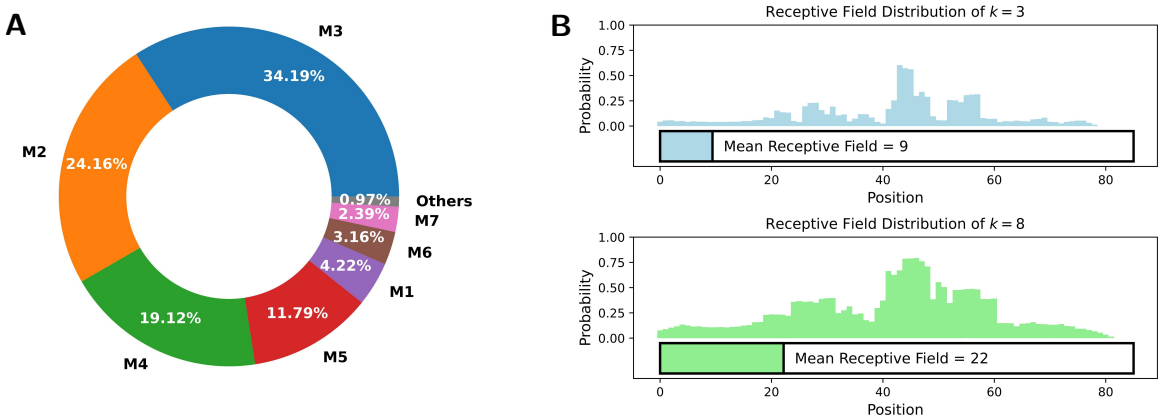

Fig. 2. **(A)** The proportion of different mutation times in Trc synthetic promoter library. Mt ($t = 1, 2, \cdots$) stands for synthetic promoters with t different mutation sites. The vast majority ($> 99\%$) of synthetic promoters have no more than 8 mutation sites. **(B)** The probability that each 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 9 and 22, 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, *Enhance* reflects the effect improvement of EVMP models compared to non-EVMP models, measured as the reduction in MAE. All MAEs are the results of 5-fold cross validation and are expressed as mean ± standard deviation.

Method	Model	Mutant	k	Valid MAE	Enhance	Test MAE	Enhance	SOTA
non-EVMP	SVM			0.2522 ± 0.0050		0.2604 ± 0.0012		0.2262
	GBDT(Zhao et al., 2021)			0.2357 ± 0.0096		0.2434 ± 0.0042		
	XGBoost(Zhao et al., 2021)			0.2193 ± 0.0107		0.2340 ± 0.0018		
	RF(Zhao et al., 2021)			0.2273 ± 0.0098		0.2262 ± 0.0022		
	LSTM(Zhao et al., 2021)			0.4959 ± 0.0213		0.5337 ± 0.0011		
	Transformer			0.2744 ± 0.0171		0.2912 ± 0.0020		
EVMP	SVM	Mask	8	0.2337 ± 0.0060	7.32%	0.2414 ± 0.0012	7.27%	**0.1917**
	GBDT			0.2197 ± 0.0096	6.78%	0.2285 ± 0.0030	6.14%	
	XGBoost			0.2051 ± 0.0143	6.45%	0.2183 ± 0.0036	6.72%	
	RF			0.2231 ± 0.0132	1.83%	0.2145 ± 0.0012	5.15%	
	LSTM	Vars+PE	3	0.1976 ± 0.0155	**60.15%**	0.2065 ± 0.0065	**61.30%**	
	Transformer		5	**0.1839 ± 0.0126**	**32.98%**	**0.1917 ± 0.0013**	**34.17%**	
EVMP SOTA Enhance								**15.25%**

a 9:1 training set and validation set in each cross-validation. All experiments used 5-fold cross-validation. It should be noted that our results could not be directly compared to those in Zhao et al. (2021) for fairness since the proportion of the training set was reduced and different independent test sets were adopted. For a fair comparison, the competitive models in Zhao et al. (2021) were all retrained under the same conditions, and we adopted the source code of the original paper, located at https://github.com/YuDengLAB/Predictive-the-correlation-between-promoter-base-and-intensity-through-models-comparing. The results of both EVMP and non-EVMP models for the synthetic promoter strength prediction task were presented in Table 1, and the evaluation metric used is MAE (mean absolute error).

3.2. EVMP Discards Redundant Bases

In the synthetic promoter library, most of the bases between different promoters are identical, which limits the models' ability to gain informative features from these shared 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.

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. As depicted in Figure 2.B, when $k = 3$ and 8, the mean receptive fields of VarEncoder turn out to be 9 and 22 respectively, corresponding to 10.6% and 25.9% of the entire promoter length. The VarEncoder receives inputs that discard redundant bases while retaining necessary mutation information, making the EVMP approach distinct from traditional NLP models.

3.3. EVMP Enhances the Effect of ML Models

The results of the comparison between the effects of EVMP and non-EVMP models are presented in Table 1. Among the conventional machine learning models, the state-of-the-art (SOTA) non-EVMP model was RF (0.2262), while the

Table 2. Results of EVMP-Transformer ($k = 5$) in synthetic promoter strength prediction task with data augmentation. The entries *EVMP* and *DA* indicate whether to use EVMP and whether to use data augmentation, respectively. The entries *Train/Valid* and *Test* refer to the dataset processing method, where *Fixed* 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 *Augmented* refers to generating 10 copies of data by using all 10 base promoters. All MAEs are the results of cross validation.

EVMP	DA	DA Method		Valid MAE	Test MAE
		Train/Valid	Test		
×	×	Fixed	Fixed	0.2744 ± 0.0171	0.2912 ± 0.0020
✓	×	Fixed	Fixed	0.1839 ± 0.0126	0.1917 ± 0.0013
✓	×	Rand	Rand	0.2423 ± 0.0076	0.2396 ± 0.0142
✓	✓	Augmented	Augmented	**0.1824 ± 0.0176**	**0.1856 ± 0.0008**
EVMP+DA SOTA Enhance					**17.95%**

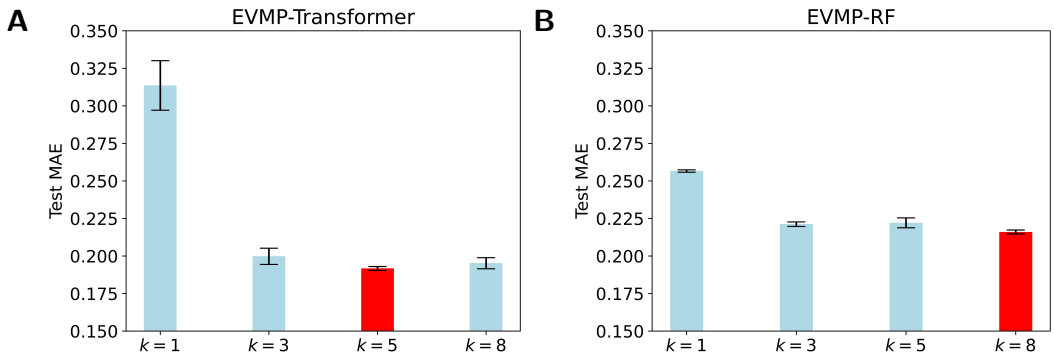

Fig. 3. Test MAE of EVMP models with different $k = 1, 3, 5, 8$ (length of k-mer Mutations). As k increased, the effect of models first improved and then tended to be the same. All MAEs are the results of cross validation. **(A)** Test MAE diagrams of EVMP-Transformer with $k = 1, 3, 5, 8, 10$, $k = 5$ obtained the best result. **(B)** Test MAE diagrams of EVMP-RF, $k = 8$ obtained the best result.

MAE of the other non-EVMP models ranged from 0.23 to 0.26. However, the performance of non-EVMP LSTM (0.5337) and Transformer (0.2912) models was inferior to that of the other ML models.

After applying the EVMP method, all models demonstrated varying degrees of improvement. The LSTM (0.5337 → 0.2065) and Transformer (0.2912 → 0.1917) showed a 61.30% and 34.17% improvement, respectively, surpassing the performance of the SOTA non-EVMP model (0.2262) and setting a new SOTA record. Other models also showed improvement ranging from 5.15% to 7.27%. The application of EVMP enhanced the performance of each model in predicting synthetic promoter strength, with the SOTA EVMP model (EVMP-Transformer, 0.1917) outperforming the SOTA non-EVMP model (RF, 0.2262) by 15.25%.

3.4. Data Augmentation Further Enhances EVMP

In our previous discussion, we did not incorporate optional data augmentation in the EVMP process. As described in Section 2.1.2, data augmentation involves selecting multiple base promoters $(P_1, P_2, \cdots, P_{10})$ and generating multiple copies of data $\langle P_1, M(P_1, S')\rangle, \langle P_2, M(P_2, S')\rangle, \cdots, \langle P_{10}, M(P_{10}, S')\rangle$ for each synthetic promoter S'. It is worth noting that the k-mer mutations $M(P_i, S')$ differ based on the selection of the base promoter $P_i, i = 1, \cdots, 10$. The ten selected base promoters, which differ from P_{trc} by 2 to 7 different sites, are listed in Table A4 in Appendix 2.

Table 2 presents the relevant comparative experiments of data augmentation. In these experiments, we chose EVMP-Transformer ($k = 5$) model for analysis. The first two Fixed-Fixed experiments were the same as in Table 1. The third Rand-Rand experiment randomly selected one of the 10 base promoters $P_1, \cdots, P_{10}$ as the base promoter, and each synthetic promoter may had a different base promoter. The fourth Augmented-Augmented experiment used all 10 base promoters and expanded the dataset by a factor of 10 by applying data augmentation.

Compared with the Fixed-Fixed method, the EVMP Rand-Rand method (0.2396) exhibited weaker performance than the EVMP Fixed-Fixed method (0.1917), but still outperformed the non-EVMP Fixed-Fixed method (0.2912). It is important to note that the Rand-Rand method represented an extreme scenario, where only approximately 1/10 of the original dataset was utilized by directly using P_{trc} as the base promoter. Nevertheless, the Rand-Rand method still achieved better performance than the non-EVMP method, which highlights the robustness of EVMP.

The Augmented-Augmented method with data augmentation (EVMP+DA) achieved the best performance (0.1856) among all experiments, which was 17.95% better than the non-EVMP SOTA. After data augmentation, multiple copies of data in the EVMP format were generated for each synthetic promoter from different perspectives (base promoters). In addition, the Augmented-Fixed method (not listed in Table 2), achieved a MAE of 0.1876 on the same Fixed (P_{trc}-based) test set in the main experiment, which was better than the EVMP method (0.1917) and lower than the average effect of all base promoters

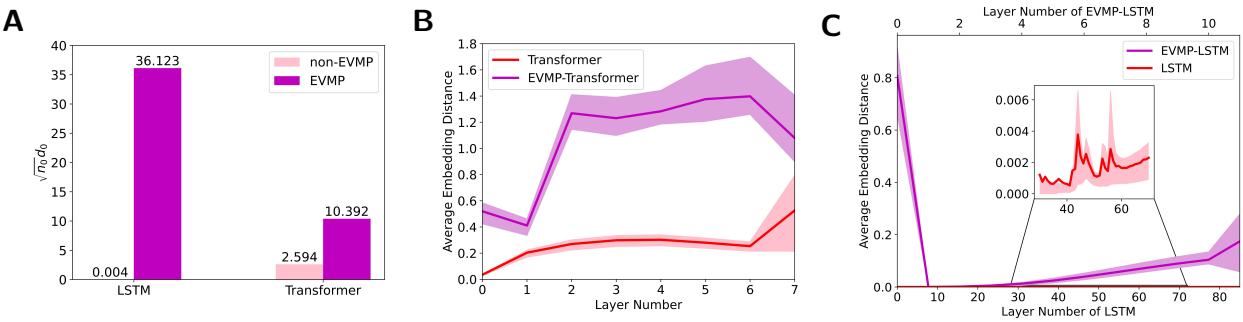

Fig. 4. Average embedding distance for each layer of deep learning models, with the 0-th layer serving as the input. Since BaseEncoder receives the same base promoter P_{trc}, non-EVMP models calculate the average embedding distance of each layer directly, while EVMP models calculate the average embedding distance of VarEncoder. **(A)** non-norm d_0 ($\sqrt{n_0}d_0$) for 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.

(0.1856). These results suggest that data augmentation indeed increased the diversity of the data to improve the model's performance, and was able to identify a better base promoter than P_{trc}. Therefore, data augmentation is a simple and effective alternative to rationally selecting base promoters.

3.5. Extended Vision is Necessary for EVMP

Section 2.2 emphasized the crucial importance of extended vision for the effectiveness of EVMP. The inclusion of k-mer mutations with $k > 1$ achieves this extended vision. Although it may seem that only including single base mutations ($k = 1$) in the mutation input is the most straightforward approach, a single base may not provide sufficient information for the promoter. As a result, many biological software programs, such as Allesøe et al. (2021) and Nurk et al. (2017), use k-mers. Experimental results in Figures 3.A and 3.B, which show the Test MAE of EVMP-Transformer and EVMP-RF for different values of $k = 1, 3, 5, 8$, support this claim.

The Test MAE of EVMP-Transformer with $k = 5$ was the best with a value of 0.1917, compared to 0.3136 for $k = 1$, 0.1998 for $k = 3$, 0.1952 for $k = 8$. The Test MAE of EVMP-RF with $k = 8$ was the best with a value of 0.2160, compared to 0.2566 for $k = 1$, 0.2212 for $k = 3$, 0.2221 for $k = 5$. These results indicate that $k > 1$ is much better than $k = 1$ and that extended vision is an indispensable key to the success of EVMP. Furthermore, the effect of the models did not change significantly as k increased. Therefore, choosing a suitable k is not difficult, and the key is to ensure that $k > 1$. To avoid excessive overlap between k-mer mutations, we suggest using the formula $k = l/\alpha$, where l is the length of the promoter and α is the average number of mutation sites.

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 (Ruderman et al., 2018). However, considering the fact that the differences between synthetic promoters are much shallower, 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 particular, a higher average embedding distance indicates a lighter over-smoothing phenomenon.

In Figure 4, the average embedding distance of each layer was calculated for EVMP and non-EVMP implementations of 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 offers a comparison of the average embedding distance among layers of neural networks. The purpose is to verify intuitively whether EVMP mitigates the over-smoothing phenomenon and to elucidate the effectiveness of EVMP. In this context, a theoretical illustration is presented to support the validity of this perspective.

Zou et al. (2020) 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 every pair of the training samples.

Here, we regard $\sqrt{n_0}d_0$ as an approximation of ϕ, where n_0 is the dimension of input. As shown in Figure 4.A, EVMP resulted in a noticeable increase in ϕ for deep learning models.

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 (Zou et al., 2020). The above discussion shows that EVMP can indeed alleviate the over-smoothing phenomenon, which contributes to its effectiveness.

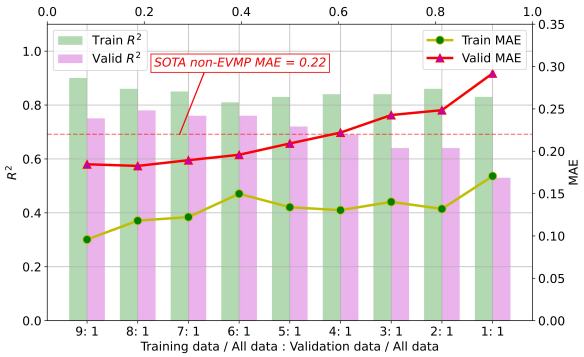

Fig. 5. As the proportion of the training set decreased, the MAE of the validation set increased and the R^2 decreased. When the proportion of training set to validation set is higher than 5:1, EVMP-Transformer achieved lower MAE than SOTA non-EVMP models.

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 (coefficient of determination) of training and validation sets for each EVMP-Transformer model.

According to our experimental results in Table 1, the SOTA non-EVMP MAE was approximately 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 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 enhanced many models and achieved 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 verified that extended vision, or k-mer, is critical for the effectiveness of EVMP. In terms of interpretability, EVMP is proved to be able to alleviate the over-smoothing phenomenon and thus improves the effect of models.

EVMP is a highly robust and versatile machine learning enhancement framework with the potential to be extended to various mutation-based synthetic biology component libraries. In future research, other ways of mutation representation and more efficient information transfer between BaseEncoder and VarEncoder can be further explored.

Supplementary Material

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

Funding

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

Data Availability

The source code is available at https://github.com/Tiny-Snow/EVMP, including all non-EVMP and EVMP models implemented in the paper.

References

Rosa L Allesøe, Camilla K Lemvigh, My VT Phan, Philip TLC Clausen, Alfred F Florensa, Marion PG Koopmans, Ole Lund, and Matthew Cotten. Automated download and clean-up of family-specific databases for kmer-based virus identification. *Bioinformatics*, 37(5):705–710, 2021.

Hal Alper, Curt Fischer, Elke Nevoigt, and Gregory Stephanopoulos. Tuning genetic control through promoter engineering. *Proceedings of the National Academy of Sciences*, 102(36):12678–12683, 2005.

Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Andrew P Cazier and John Blazeck. Advances in promoter engineering: novel applications and predefined transcriptional control. *Biotechnology Journal*, 16(10): 2100239, 2021.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Josh T Cuperus, Benjamin Groves, Anna Kuchina, Alexander B Rosenberg, Nebojsa Jojic, Stanley Fields, and Georg Seelig. Deep learning of the regulatory grammar of yeast 5' untranslated regions from 500,000 random sequences. *Genome research*, 27(12):2015–2024, 2017.

Carl G de Boer, Eeshit Dhaval Vaishnav, Ronen Sadeh, Esteban Luis Abeyta, Nir Friedman, and Aviv Regev. Deciphering eukaryotic gene-regulatory logic with 100 million random promoters. *Nature biotechnology*, 38(1):56–65, 2020.

Ronald PH de Jongh, Aalt DJ van Dijk, Mattijs K Julsing, Peter J Schaap, and Dick de Ridder. Designing eukaryotic gene expression regulation using machine learning. *Trends in biotechnology*, 38(2):191–201, 2020.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

Jucan Gao, Lihong Jiang, and Jiazhang Lian. Development of synthetic biology tools to engineer pichia pastoris as a chassis for the production of natural products. *Synthetic and systems biotechnology*, 6(2):110–119, 2021.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Pouya Kheradpour and Manolis Kellis. Systematic discovery and characterization of regulatory motifs in encode tf binding experiments. *Nucleic acids research*, 42(5):2976–2987, 2014.

Benjamin J Kotopka and Christina D Smolke. Model-driven generation of artificial yeast promoters. *Nature communications*, 11(1):1–13, 2020.

Sudhir Kumar, Masatoshi Nei, Joel Dudley, and Koichiro Tamura. MEGA: A biologist-centric software for evolutionary analysis of DNA and protein sequences. *Briefings in Bioinformatics*, 9(4):299–306, 04 2008.

Wen-li Liu and Qing-biao Wu. Analysis method and algorithm design of biological sequence problem based on generalized k-mer vector. *Applied Mathematics-A Journal of Chinese Universities*, 36(1):114–127, 2021.

R Scott McIsaac, Patrick A Gibney, Sunil S Chandran, Kirsten R Benjamin, and David Botstein. Synthetic biology tools for programming gene expression without nutritional perturbations in saccharomyces cerevisiae. *Nucleic acids research*, 42(6):e48–e48, 2014.

Sergey Nurk, Dmitry Meleshko, Anton Korobeynikov, and Pavel A Pevzner. metaspades: a new versatile metagenomic assembler. *Genome research*, 27(5):824–834, 2017.

Avraham Ruderman, Neil C Rabinowitz, Ari S Morcos, and Daniel Zoran. Pooling is neither necessary nor sufficient for appropriate deformation stability in cnns. *arXiv preprint arXiv:1804.04438*, 2018.

Hongting Tang, Yanling Wu, Jiliang Deng, Nanzhu Chen, Zhaohui Zheng, Yongjun Wei, Xiaozhou Luo, and Jay D Keasling. Promoter architecture and promoter engineering in saccharomyces cerevisiae. *Metabolites*, 10(8):320, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Matthew T Weirauch, Ally Yang, Mihai Albu, Atina G Cote, Alejandro Montenegro-Montero, Philipp Drewe, Hamed S Najafabadi, Samuel A Lambert, Ishminder Mann, Kate Cook, et al. Determination and inference of eukaryotic transcription factor sequence specificity. *Cell*, 158(6):1431–1443, 2014.

Ming-Ru Wu, Lior Nissim, Doron Stupp, Erez Pery, Adina Binder-Nissim, Karen Weisinger, Casper Enghuus, Sebastian R Palacios, Melissa Humphrey, Zhizhuo Zhang, et al. A high-throughput screening and computation platform for identifying synthetic promoters with enhanced cell-state specificity (specs). *Nature communications*, 10(1):1–10, 2019.

Mei Zhao, Zhenqi Yuan, Longtao Wu, Shenghu Zhou, and Yu Deng. Precise prediction of promoter strength based on a de novo synthetic promoter library coupled with machine learning. *ACS Synthetic Biology*, 11(1):92–102, 2021.

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine learning*, 109(3):467–492, 2020.