OXFORD

Briefings in Bioinformatics, 2024, **25(5)**, bbae398
https://doi.org/10.1093/bib/bbae398
Problem Solving Protocol

CAPE: a deep learning framework with Chaos-Attention net for Promoter Evolution

Ruohan Ren[1,‡], Hongyu Yu[2,‡], Jiahao Teng[3], Sihui Mao[1], Zixuan Bian[4], Yangtianze Tao[2], Stephen S.-T. Yau[2,5,*]

[1]Zhili College, Tsinghua University, Beijing 100084, China
[2]Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China
[3]School of Life Sciences, Tsinghua University, Beijing 100084, China
[4]Weiyang College, Tsinghua University, Beijing 100084, China
[5]Beijing Institute of Mathematical Sciences and Applications (Bimsa), Beijing 101408, China

[*]Corresponding author. Beijing Institute of Mathematical Sciences and Applications (Bimsa), Beijing 101408, China. Email: yau@uic.edu
[‡]Ruohan Ren and Hongyu Yu contributed equally to this work.

Abstract

Predicting the strength of promoters and guiding their directed evolution is a crucial task in synthetic biology. This approach significantly reduces the experimental costs in conventional promoter engineering. Previous studies employing machine learning or deep learning methods have shown some success in this task, but their outcomes were not satisfactory enough, primarily due to the neglect of evolutionary information. In this paper, we introduce the Chaos-Attention net for Promoter Evolution (CAPE) to address the limitations of existing methods. We comprehensively extract evolutionary information within promoters using merged chaos game representation and process the overall information with modified DenseNet and Transformer structures. Our model achieves state-of-the-art results on two kinds of distinct tasks related to prokaryotic promoter strength prediction. The incorporation of evolutionary information enhances the model's accuracy, with transfer learning further extending its adaptability. Furthermore, experimental results confirm CAPE's efficacy in simulating *in silico* directed evolution of promoters, marking a significant advancement in predictive modeling for prokaryotic promoter strength. Our paper also presents a user-friendly website for the practical implementation of *in silico* directed evolution on promoters. The source code implemented in this study and the instructions on accessing the website can be found in our GitHub repository https://github.com/BobYHY/CAPE.

Keywords: promoter design; directed evolution; deep learning; chaos game representation

Introduction

In the field of synthetic biology, precise characterization of regulatory elements is of paramount importance for the design of synthetic gene circuits [1, 2]. Such characterization can significantly advance many critical domains, including pharmaceutical synthesis [3, 4], metabolic engineering [5], and material production [6, 7]. Among the various regulatory elements, promoters play a pivotal role in synthetic biology [8], as they exert significant control over the expression level of downstream genes [9, 10]. Therefore, identifying a promoter with the appropriate strength is crucial for constructing expression vectors, and the optimization of promoter sequences is a key task in synthetic biology.

Conventional promoter engineering relies on experimental techniques for identifying suitable promoters, including mutagenesis [11, 12], sequence combinations [13], etc. One commonly adopted method involves random mutagenesis of promoters through error-prone Polymerase Chain Reaction (PCR), followed by the selection of mutants with increased strength [14]. This iterative process is often referred to as the directed evolution of promoters. Nevertheless, experimental methods are frequently characterized by high levels of unpredictability and labor intensiveness.

The development of artificial intelligence has created the foundation for *in silico* directed evolution of promoters, with a key prerequisite being the establishment of an accurate regression model that correlates promoter sequences with their strengths. There has been some related research on computational models for prokaryotic promoters, including machine learning or deep learning models. However, many models are used to identify whether a given sequence can serve as a promoter [15–17]. For promoter strength prediction, most of the existing models are classification models, used to predict whether a promoter is strong or weak [18–20]. Currently, there is still a lack of accurate regression models in this regard. It is worth mentioning that Wang *et al.* combined a deep generative model with a predictive model to preselect the most promising synthetic promoters [21]. This is an important pioneering work in the field of deep generative models for promoters. However, due to the difficulty of avoiding noise interference, the Pearson correlation coefficient (PCC) of their Convolutional Neural Network (CNN)-based predictive model was around 0.25, suggesting a pressing need for substantial improvement.

Understanding the evolutionary history of corresponding promoters plays a crucial role when aiming for directed promoter

Received: May 15, 2024. **Revised:** July 13, 2024. **Accepted:** July 27, 2024

evolution. The significance of evolutionary information in deep learning models has been successfully demonstrated in fields such as protein structure prediction [22] and protein-protein interaction prediction [23]. However, in the field of promoter design, suitable evolutionary information features have yet to be successfully applied. The limited availability of promoter data poses a challenge for feature extraction using traditional alignment algorithms, as finding an adequate number of suitably similar promoters proves to be difficult. Leveraging the alignment-free chaos game representation [24] allows us to extract the inherent evolutionary information within promoters with only local or moderate similarity, offering valuable support to enhance the model's effectiveness.

In this paper, we propose the Chaos-Attention net for Promoter Evolution (CAPE), which features the incorporation of merged chaos game representation and the utilization of modified DenseNet [25] and Transformer [26]. CAPE is a highly accurate regression model that establishes correlations between promoter sequences and their strengths, leading to state-of-the-art (SOTA) performance. Through the evolutionary information extracted from promoter sequences, our deep learning model achieved a PCC of 0.52 on the dataset from [27] by a five-fold cross-validation, significantly surpassing 0.24 by Wang *et al.*'s method [21] and 0.27 by the predictor of DeepSEED [28] on the same dataset with the same cross-validation. Furthermore, we implemented transfer learning to enhance the model's adaptability for other downstream tasks, for example, the strength prediction task of the trc promoters. The original trc promoter is a synthetic composite of trp and lac promoters [29, 30]. It is an important synthetic promoter, and therefore many researchers have studied the strength of its variants, resulting in relatively abundant data [31, 32]. When applied to predict the strength of trc promoters, our model achieved an R-squared (R2) value of 0.68, signifying a substantial improvement over Zhao *et al.*'s methods (0.63 for the best one) [31] and six EVMP-based algorithms (0.63 for the best one) [32]. This underscores the considerable superiority of our model structure. Finally, we conducted biological experiments on two different kinds of promoters(constitutive promoter and inducible promoter), and the results indicate that our model is indeed capable of efficiently evolving promoters.

In summary, we harnessed evolutionary information to construct a deep learning model CAPE, which enabled us to attain SOTA performance in predicting prokaryotic promoter strength. We confirmed the model's effectiveness and wide applicability in simulating the directed evolution of promoters *in silico* through biological experiments. We also developed a website for convenient implementation of directed evolution on promoters.

Materials and methods

Dataset

In our research, we utilized three datasets, which are as follows:

The first dataset, named dataset_pro, is derived from the PPD database and comprises 129 148 experimentally validated promoter sequences across 63 prokaryotic species [33]. We conducted sequence alignment within dataset_pro to identify similar promoter sequences for investigating the evolutionary history of the studied promoters in other datasets. Please note that dataset_pro only contains experimentally confirmed promoter sequences, but does not include the corresponding strengths of the promoters. Therefore, we did not design any training tasks based on this dataset. Instead, we used it as a database to search for similar sequences to the promoters we want to study.

The second dataset, dataset_Ecoli, contains 11 884 artificially defined promoter sequences of *Escherichia coli*, along with the corresponding gene expression strengths measured by dRNA-seq. Since prokaryotes do not have many regulatory elements like eukaryotic enhancers, the expression level of their corresponding genes can be indirectly regarded as the strength of the promoter. This dataset originated from Thomason *et al.* [27] and was employed by Wang *et al.* [21] for predictive model. We also used dataset_Ecoli to train our model. Please note that we refer to the task designed based on dataset_Ecoli as Task1.

The third dataset, dataset_trc, comprises 3665 mutated trc promoter sequences and their corresponding promoter strengths. This dataset, introduced by Zhao *et al.* [31], was constructed using 83 rounds of mutation-construction-screening-characterization engineering cycles. The strength of the promoters was determined by fluorescent protein intensity. We employed dataset_trc to validate the transfer learning capability of our constructed predictive model and test the predictive performance after fine-tuning. Please note that we refer to the task designed based on dataset_trc as Task2.

Overview of the model architecture

The architecture of CAPE is as follows (Fig. 1):

First, we employed Basic Local Alignment Search Tool (BLAST) [34] and the Needleman–Wunsch (NW) algorithm [35] to search for sequences exhibiting a certain level of similarity with the target promoter within a prokaryotic promoter database [33]. Subsequently, we applied a novel method firstly introduced in this paper, referred to as merged CGR, to convert the promoter sequence into image data capturing evolutionary information. Alongside image information, we applied the kmer2vec method [36] to extract textual information from the promoter sequences.

The above two types of information will be input into two different deep learning networks, namely DenseNet [25] and Transformer [26], respectively. We adapted the structure of DenseNet and Transformer to suit our tasks. The results processed by both models are fed into a fully connected network for integration. Finally, our model can output the predicted strengths of given promoter sequences. Moreover, we introduced a fine-tuning network for transfer learning, which enhances the model's ability to adapt to various downstream tasks.

Merged chaos game representation

Merged Chaos Game Representation, abbreviated as Merged CGR, is a novel feature extraction method proposed for the first time in this paper. This approach is built upon the conventional Chaos Game Representation [24] and extends its applicability to not only the given sequence but also its related sequences, converting them into a unified matrix (Fig. 2a). Subsequently, this matrix is input into the DenseNet [25] for further processing. Notably, this representation method functions as an alternative to the widely employed position-specific scoring matrix (PSSM) found in other related studies.

For a sequence $s = (s_1 s_2 \ldots s_n)$, generating the corresponding Merged CGR involves three steps. The first step is to transform the sequence into a matrix using conventional CGR. The CGR sequence corresponding to s, $X_i = (x_i, y_i)$ where $i = 1, \ldots, n$, is given by

$$X_0 = \left(\frac{1}{2}, \frac{1}{2}\right), \ X_i = \frac{1}{2}\left(X_{i-1} + W_i\right) \tag{1}$$

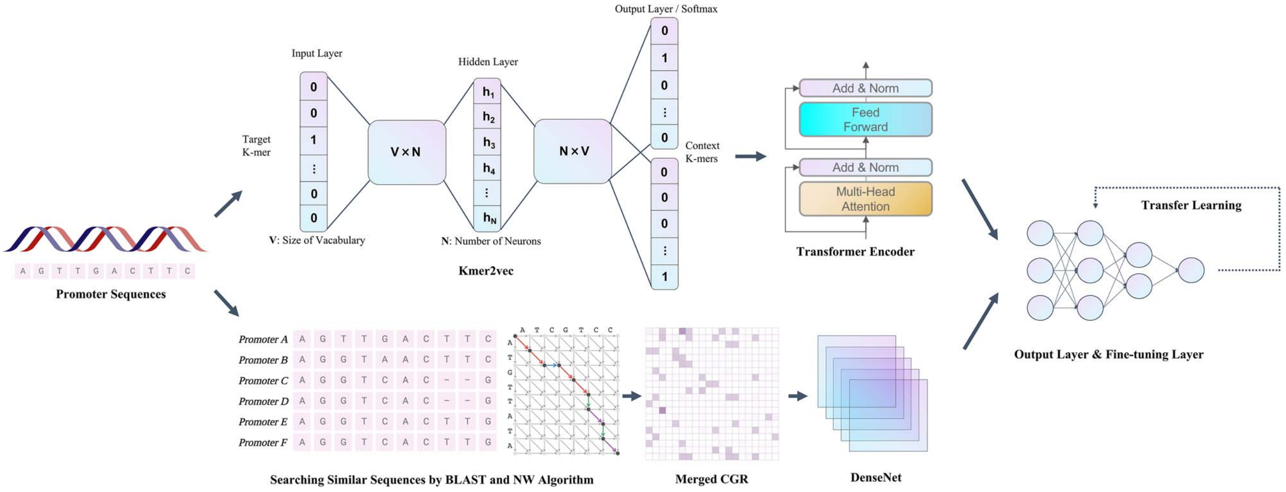


Figure 1. Overview of the model architecture.

where W_i equals to $(0, 0)$, $(1, 0)$, $(0, 1)$, $(1, 1)$ if s_i is A, T, C, G respectively. By uniformly subdividing the unit square into L^2 subsquares and calculating the number of points within each subdivision, the CGR sequence can be further transformed into a CGR matrix with L rows and L columns, denoted as $CGR(s)$ ($L = 20$). The second step involves using BLAST [34] (blastn-short, evalue = 1) to search for matching sequences $s^{(1)}, ..., s^{(m)}$ in dataset_pro. Subsequently, the NW algorithm [35] is applied to confirm their similarity to s. The similarity is calculated as the sum of the scores in the pairing, where both mismatch and gap have a score of -1, and match has a score of 1. This sum is then divided by the promoter length of 50, resulting in similarity scores $a_1, ..., a_m$ ($a_i \leq 1$). Finally, considering all sequences with similarity scores greater than 0, the Merged CGR matrix is computed as $CGR(s) + \sum_{i=1}^{m} a_i CGR(s^{(i)}) 1_{\{a_i>0\}}$.

We have chosen to apply Merged CGR instead of PSSM to extract the evolutionary information for promoters for the following reason. We aim to restrict our search for related sequences to the promoter region. Considering the relatively limited availability of promoter data, which frequently includes orphan promoters, it becomes challenging to identify highly similar sequences suitable for PSSM generation. In contrast, our Merged CGR method excels in integrating sequences exhibiting moderate or localized similarity, thereby accommodating such orphan promoter scenarios.

Word2vec word embedding

DNA sequences can be divided into a series of k-mers [37–39], allowing the sequence to be treated as text where the k-mers serve as words. Accordingly, we can use word embedding techniques from natural language processing (NLP) to represent these k-mers numerically.

The word2vec method, proposed by Mikolov *et al.* [40, 41], embeds words into meaningful high-dimensional numerical vectors. The neural network structure includes the continuous bag-of-word (CBOW) model or the skip-gram model. During training, CBOW mainly predicts a word from its context, while skip-gram predicts the context words, given a certain word. For instance, given a sequence of training words $w_1, w_2, w_3, \ldots, w_T$, the objective of the skip-gram model is to maximize the average log probability described as follows:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j} \mid w_t) \tag{2}$$

where c is the size of the training context (which can be a function of the center word w_t). The probability is defined as:

$$p(w_O|w_I) = \frac{\exp\left((v'_{w_O})^\top v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left((v'_w)^\top v_{w_I}\right)} \tag{3}$$

where v_w and v'_w represent the ‘input’ and ‘output’ vector representations of w, respectively, and W is the total number of words in the vocabulary. Each word’s vector representation during training is influenced by its surrounding vocabulary. If two words have similar contextual vocabulary, their word vectors will also be similar.

To better extract textual information from promoter sequences, we employed the word2vec method to obtain k-mer word embeddings in promoter sequences, following the specific strategies of the kmer2vec method [36]. Initially, we divided all promoter sequences in dataset_pro into a series of 3-mers using an overlapping division, treating them as complete text. Subsequently, we conducted training (window_size = 24; vector_size = 100; the skip-gram algorithm) on the text created from dataset_pro. This method allows us to leverage the kmer2vec approach for representing k-mer sequences in DNA, capturing both sequence similarity and functional characteristics within the evolutionary context of promoters.

DenseNet

DenseNet (Densely Connected Convolutional Network) [25] is a deep learning algorithm designed for processing matrix data. It serves as an extension of traditional CNNs, with its most prominent feature being the dense connections between layers. These connections enable comprehensive information exchange among layers, allowing shallow-level information to be retained even after passing through multiple layers. This, in turn, allows for the training of deeper networks. Unlike ResNet [42], where connections involve addition, DenseNet employs a different approach by concatenating data across the channel dimension. Specifically, for the lth layer, it takes the outputs of all preceding l-1 layers as inputs, that is,

$$x_l = H_l([x_1, ..., x_{l-1}]) \tag{4}$$

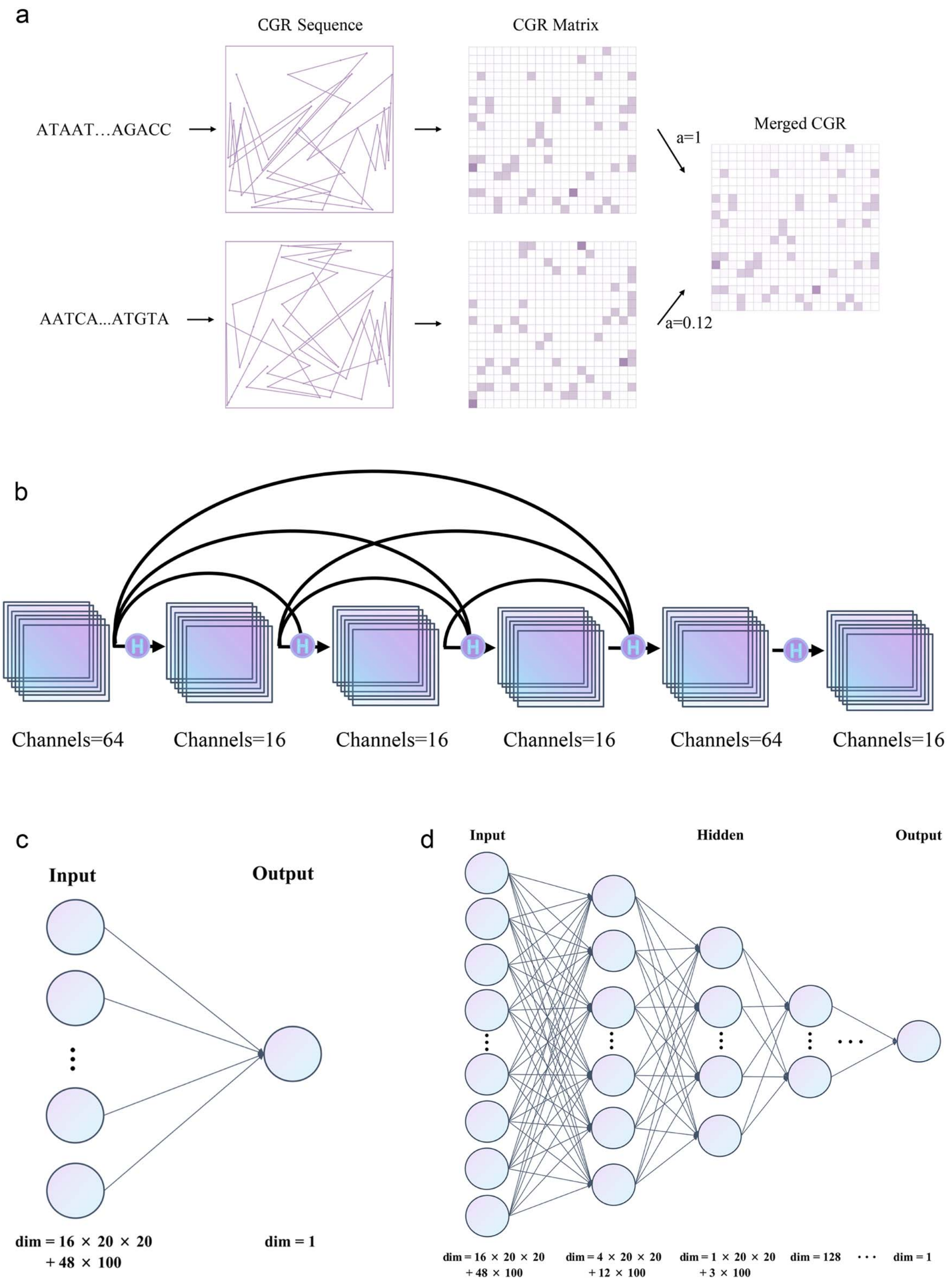


Figure 2. Modules of the model. (a) Merged CGR. (b) DenseNet. (c) The original fully connected network. (d) The fully connected network for fine-tuning.

where H_l is the function corresponding to the lth layer. The channel output of H_l assumes a crucial role in DenseNet, representing the growth rate of channels within the network (growth rate = 16 in this paper). Due to limitations in the data size, we do not employ the deep network structure as in the original work. Instead, we embed a relatively shallow network into the DenseNet architecture, as illustrated in Fig. 2(b). Each layer in this study comprises four fundamental components. In addition to the standard components commonly found in CNNs, including the convolution part, Batch Normalization, and the activation part (ReLU), an additional dropout component [43] has been introduced to address overfitting.

Transformer

The Transformer [26] is a deep learning algorithm extensively employed in processing sequential data. Its core concept is the attention mechanism. Specifically, each element in sequential data, such as words, is transformed into vectors using techniques like word embedding. These vectors are further mapped to corresponding query (Q), key (K), and value (V) through various linear transformations. This allows each word to establish connections with all other words, as defined by the following equation:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

where d_k is the dimension of Q and K. This attention mechanism is more capable of capturing longer-range information compared to traditional RNN-based algorithms [44, 45]. Furthermore, the Transformer employs a multi-head strategy, linearly projecting the Q, K, V multiple times with different learned linear projections, enhancing the richness of information capture. In other words, we have

$$\begin{aligned} MultiHead(Q, K, V) &= Concat(head_1, \ldots, head_h)W^O \\ head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{6}$$

where W^O is also a learned parameter matrix.

Transformer employs an encoder-decoder structure, with each part consisting of multiple Transformer blocks. Within each block, the Transformer combines the attention mechanism with residual networks, transforming the output into the input for the next block. In this paper, because we do not need to convert sequences into sequences, but only into real numbers, we have exclusively employed the encoder. The encoder consists of two Transformer blocks, with the number of attention heads set to 8.

Fully connected network

The fully connected network is utilized to combine the graphical evolutionary information processed by DenseNet and the sequential linguistic information processed by Transformer. A single-layer network suffices for the original model (Fig. 2c). Throughout the fine-tuning process, we maintain the parameters of the preceding model as existing knowledge and solely replace the fully connected network for retraining. Therefore, to enhance flexibility, multiple layers of fully connected networks are employed (Fig. 2d).

Directed evolution of promoters

Using our model to perform promoter directed evolution *in silico* requires the following two steps.

Step 1 (Random Mutation): First, we should choose the target promoter and its evolution direction. For example, in our project, we intended to evolve the promoter PnisA to have a stronger strength. If the sequence length is largar than 50 bp, we only consider the last 50 bp. We then used a program to randomly mutate the sequence of this promoter, and the number of mutated bases can be adjusted (usually ≤ 5). We can then obtain a large number of mutated sequences on the basis of the original promoter sequence.

Step 2 (Strength Screening): We input the large number of mutant sequences into our model and selected the top sequences with the highest predicted strength based on the model feedback. In this way, we completed the directed evolution of the promoter *in silico*, which can then be tested experimentally.

Bacterial strains and cultivation

The *E. coli Stbl3* strain was used for plasmid cloning and fluorescent protein expression analysis. Inocula were cultured in 5 mL of Luria Bertani (LB) medium containing antibiotics (kanamycin for the pET28A plasmid and carbenicillin for the pBV220 plasmid) in 15 mL shaking culture tubes at 200 rpm, 37°C. For gene expression analysis, with respect to constitutive promoters, we inoculated 50 μL of bacterial culture into 1 mL of LB medium containing antibiotics and incubated at 37°C with shaking at 200 rpm for 12-14 hours to ensure bacterial growth in the logarithmic phase. For the heat-inducible PL promoter, we initially inoculated 50 μL of bacterial culture into 1 mL of LB medium containing antibiotics and incubated at 37°C with shaking at 200 rpm for 4–6 hours. Subsequently, cells were induced expression by placing the culture in a 42°C water bath for 1 hour and then returned it to 37°C with shaking at 200 rpm for 10–12 hours. All strains and plasmids are listed in Supplementary Table 1.

Plasmid construction

The PnisA promoter was ligated into the pET28a-GST-mCherry vector. Specific promoter mutation primers were designed for each promoter (Supplementary Table 2). PCR amplification was performed using Q5® High-Fidelity DNA Polymerase (NEB #M0491L). The original vector was digested with the restriction enzyme DpnI (NEB #R0176S) for 4 hours at 37°C. Subsequently, the amplified PCR fragments were separated by 1.5% agarose gel electrophoresis, and gel purification was carried out using the NucleoSpin.Gel and PCR clean-up kit (MN #740609.25). The PCR fragments and linearized pET28a-GST-mCherry vector were then subjected to homologous recombination using the NEBuilder® HiFi DNA Assembly Master Mix (NEB #E2621L) to obtain the ligated product. For the PL promoter, specific promoter mutation primers were designed (Supplementary Table 2). Q5® High-Fidelity DNA Polymerase (NEB #M0491L) was used for circular PCR on the pBV220-mCherry vector, resulting in the pBV220-PL-mCherry plasmid. The original vector was digested with the restriction enzyme DpnI (NEB #R0176S) for 4 hours at 37°C. The ligated products obtained were transformed into *E. coli Stbl3* strain using the heat shock method. The sequence accuracy was confirmed through Sanger sequencing.

Fluorescence expression assay

For constitutive promoter PnisA, 50 μL of bacterial culture was inoculated at a 1:50 dilution into 2.5-mL LB medium containing kanamycin, and the culture was placed on a shaker at 200 rpm and 37°C for 14 hours to allow the cells to enter the logarithmic growth phase. For the temperature-inducible PL promoter, after reaching the logarithmic growth phase, the culture was induced using the method described above, followed by incubation on a shaker at 200 rpm and 37°C for an additional 10–12 hours. Subsequently, 0.5 mL of the bacterial culture was centrifuged at 4000 rpm for 5 minutes, the supernatant was discarded, and the pellet was resuspended in 1 mL of PBS buffer. After another centrifugation at 4000 rpm for 5 minutes and discarding the supernatant, the pellet was resuspended in 0.2 mL of PBS. The mCherry fluorescence and OD600 were then measured using a microplate reader.

Flow cytometry

E. coli in the logarithmic growth phase were harvested, centrifuged at 4000 rpm for 5 minutes, and resuspended in PBS buffer. Flow cytometry data were collected using BD Fortessa or Thermo Fisher

Attune NxT flow cytometers and analyzed with FlowJo software (BD Biosciences).

Results

Performance on two prediction tasks

To validate the effectiveness of CAPE, we conducted tests for two prediction tasks. The dataset for Task1, named dataset_Ecoli, comprises 11 884 promoter sequences along with their corresponding strengths. Each promoter is artificially defined as the 50 bp sequence preceding the transcription start site. These promoter sequences exhibit significant diversity, corresponding to different genes, allowing the use of dataset_Ecoli for the model's general training. By using a five-fold cross-validation, our model achieved an average PCC of 0.52 and an average SCC (Spearman Correlation Coefficient) of 0.39 in Task1, surpassing (PCC = 0.24, SCC = 0.21) by Wang *et al.*'s method [21] and (PCC = 0.27, SCC = 0.20) produced by the predictor of DeepSEED [28]. Our model's performance is about 2 times higher than that of the previous best-performing model (Fig. 3e). The model's substantial enhancement is impressive, significantly improving the capacity to extract information and avoid extensive noise. In Fig. 3(a), we present the scatter plot of the model's predictions on the training and test sets in a fold. In Fig. 3(c), we show that our split is random enough, resulting in a reasonably consistent distribution of promoter strengths between the training and test sets.

In synthetic biology, *Escherichia coli* stands as an indisputable model organism in prokaryotes, and many other prokaryotic promoters demonstrate functionality in *E. coli*. Therefore, the model trained on dataset_Ecoli can aid in predicting prokaryotic promoter strength. Moreover, it can simulate mutational screening and directed evolution based on the predicted strengths.

As the strengths of promoters in dataset_Ecoli were measured by dRNA-seq, this measurement method might introduce some noise. Currently, most promoter strength tests in experiments use fluorescent proteins (such as green fluorescent protein or monomeric Cherry fluorescent protein) as downstream reporters. We sought to understand whether our model is suitable for datasets using fluorescent protein strength as the promoter strength. Hence, we conducted Task2 test. We selected dataset_trc, containing 3665 promoter sequences along with their corresponding strengths. The trc promoter is a synthetic composite of trp and lac promoters [29, 30]. Though it can be induced, Zhao *et al.*, [31] considered the trc promoter as a constitutive promoter for strength screening (by removing lacI). These 3665 promoters are variant strains of the original trc promoter, displaying different strengths. Notably, Task2 differs significantly from Task1. The promoter sequences in Task1 dataset show considerable diversity, while those in Task2 have minor differences, demanding higher precision from the model in recognizing these distinctions. In addition, dataset_trc utilizes fluorescent protein strength to represent promoter strength, likely resulting in less noise.

To assess the model's transfer learning capability, as the promoter length in Task1 was 50 bp, we needed to maintain consistency between both tasks. Here, we characterized the last 50 bp of the 74-bp trc promoter sequences in the dataset, as bases closer to the transcription start site typically have a more substantial influence. In pre-training, we transferred the model obtained from Task1, fixing all parameters before the fully connected network, as we believed these parameters contained a significant amount of information regarding prokaryotic promoters. Subsequently, we employed a pyramid-shaped fully connected layer as the fine-tuning network, replacing the original fully connected network. Only the parameters of the fine-tuning layer were modified during training. Similar to all other research in dealing with dataset_trc, we choose R2 score to be the main evaluation metric to compare the results. By using a five-fold cross-validation, our model achieved an average R2 score of 0.68. By the same cross-validation, Zhao *et al.*'s methods' [31] R2 values range from 0.38 to 0.63 and six EVMP-based algorithms' [32] R2 values range from 0.56 to 0.63. This demonstrates that our model achieved an overall improvement of 8.0% compared to the previously best-performing model, showcasing exceptional transfer learning capability. To better compare these methods, we also included PCC, Spearman correlation coefficient (SCC), mean squared error (MSE), and mean absolute error (MAE) as auxiliary metrics in Table 1. The visualization of the table is shown in Fig. 3(f) by a radar chart. The scatter plot of the model's predictions on the training and test sets as well as the violin plot for the random split in a fold are displayed in Fig. 3(b) and (d), respectively.

In addition, we found that if the same model is used without adopting the transfer learning approach, i.e. without leveraging the knowledge learned from dataset_Ecoli, the R2 score would be 0.65. This result indicates that our model outperforms other models even without transferring knowledge from other datasets. It also suggests that although dataset_Ecoli and dataset_trc differ significantly in experimental types and subjects, the information embedded in the former still contributes to the modeling of the latter. In summary, we validated the model's transfer learning ability, showcasing that our model could utilize general information from Task1 to make more specific predictions such as Task2.

To better understand the role of each part of our model CAPE, we also conducted ablation experiments. Since CAPE employs an overall process including general training and fine-tuning, to comprehensively assess the roles of each module, we used the results after final fine-tuning for testing. From the data in Table 2, it can be observed that when considering all modules together (complete CAPE), the model achieves an R2 of 0.68 for fine-tuning Task2. If we remove the image information part (Merged CGR + DenseNet) and only retain the textual information part (kmer2vec + Transformer), the model achieves an R2 of 0.61 for Task2. Conversely, if we remove the textual information part (kmer2vec + Transformer) and only retain the image information part (Merged CGR + DenseNet), the model achieves an R2 of 0.65 for Task2. This indicates that both parts play important roles in the model's effectiveness, with the introduction of evolutionary information in the image information part being relatively more crucial, which aligns perfectly with our previous speculation.

Experimental results

To further validate the accuracy of CAPE, we proceeded to conduct verification experiments in the *E. coli Stbl3* strain, utilizing constitutive promoter PnisA, as well as the heat-inducible promoter PL. Please note that the PnisA promoter is an nisin-induced promoter in *Lactococcus lactis*. However, due to its baseline expression after introduction into *E. coli*, it is considered as a constitutive promoter in our research. The flowchart of the experimental process is shown in Fig. 4(a). We used the original sequences as input to generate top-performing sequences through *in silico* directed evolution for each promoter. These sequences are detailed in Supplementary Table 2. Subsequently, we successfully employed molecular cloning techniques to construct the mutated promoters alongside the gene of mCherry fluorescent protein into pET28A and pBV220 plasmid vectors. The experimental results

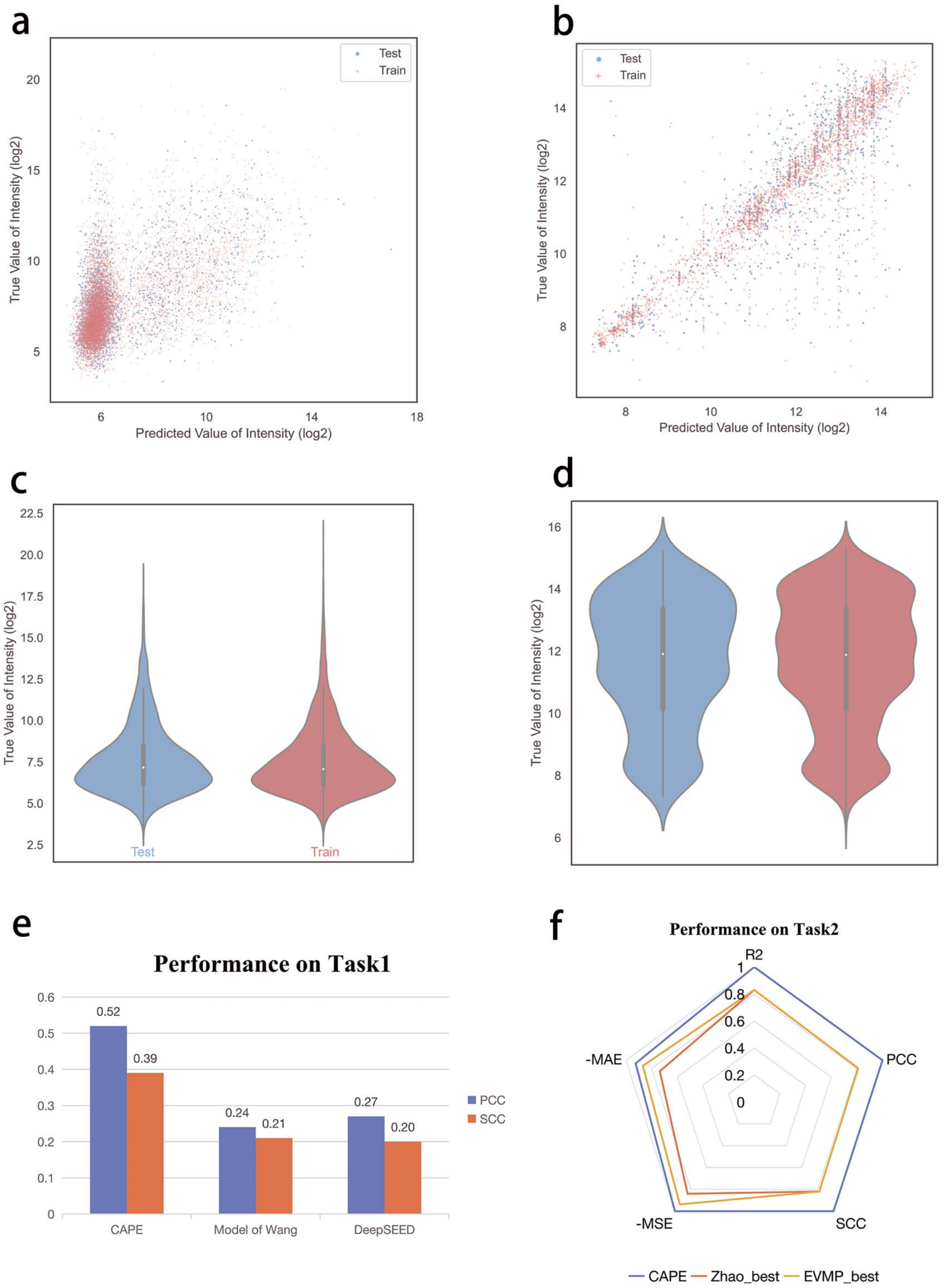


Figure 3. Performance of the model. (a) The scatter plot for model prediction on Task1. (b) The scatter plot for model prediction on Task2. (c) The violin plot for true promoter strengths on Task1 in a fold. (d) The violin plot for true promoter strengths on Task2 in a fold. (e) Average PCC and SCC for the test set on Task1. In Task1, previous models are compared through PCC, so we used PCC as well as SCC, the correlation coefficient used for nonlinear cases, to measure the methods to maintain consistency. PCC: Pearson correlation coefficient, SCC: Spearman correlation coefficient. (f) Radar chart based on five metrics in Task2. We linearly normalize the values from Table 1 by setting the best score for each metric to 1 and the worst score to 0. For clarity in visualization, we only display CAPE as well as the best methods from Zhao *et al.*'s methods' series and EVMP methods' series.

indicate that, fluorescence intensity analysis demonstrated a significant increase in promoter strength for the PnisA promoter. The highest fluorescence/OD600 (defined as unit fluorescence intensity), in comparison to the original sequence, increased to 234% (Fig. 4b). Furthermore, 37.5% of the mutated promoters demonstrated enhanced expression strength when compared to the original promoter (Fig. 4b). We also get similar results by single-cell fluorescence intensity detection in flow cytometry, and half of the mutated promoters (4/8) showed mCherry expression levels exceeding those of the original promoter, with the highest

Table 1. Model performance comparison on Task2. R2: R-squared, PCC: Pearson correlation coefficient, SCC: Spearman correlation coefficient, MSE: mean squared error, MAE: mean absolute error. RF: Random Forest, Adaboost: Adaptive Boosting, XGBoost: eXtreme Gradient Boosting, GDBT: Gradient Boosting Decision Trees, SVM: Support Vector Machine, LSTM: Long Short-Term Memory

Model	R2	PCC	SCC	MSE	MAE
CAPE	**0.68**	**0.83**	**0.81**	**1.39**	0.79
CAPE (no pre-training)	0.65	0.81	0.80	1.52	0.82
Zhao RF	0.59	0.77	0.74	1.77	0.91
Zhao Adaboost	0.38	0.66	0.63	2.67	1.28
Zhao Xgboost	0.63	0.80	0.78	1.59	0.89
Zhao GDBT	0.62	0.79	0.78	1.66	0.94
EVMP RF	0.60	0.77	0.75	1.60	0.86
EVMP Xgboost	0.63	0.79	0.78	1.47	0.82
EVMP GBDT	0.62	0.79	0.77	1.50	0.85
EVMP SVM	0.57	0.77	0.75	1.69	0.88
EVMP LSTM	0.56	0.81	0.79	1.49	**0.75**
EVMP Transformer	0.57	0.81	0.79	1.60	0.76

Table 2. Ablation experiments. R2: R-squared, PCC: Pearson correlation coefficient, SCC: Spearman correlation coefficient, MSE: mean squared error, MAE: mean absolute error

Model	R2	PCC	SCC	MSE	MAE
CAPE	**0.68**	**0.83**	**0.81**	**1.39**	**0.79**
CAPE (no image way)	0.61	0.79	0.76	1.67	0.90
CAPE (no textual way)	0.65	0.81	0.79	1.51	0.85

Median Fluorescence Intensity (MFI) surpassing the original promoter to 443% (Fig. 4b).

In addition to the constitutive promoters, we also obtained satisfactory results for the heat-inducible PL promoter. The induced fluorescence per OD600, compared to the wild-type, showed a maximum 497% increase, with 35.7% of the mutated promoters surpassing the wild-type expression level (Fig. 4c). During flow cytometry detection, 57.1% of the mutant PL promoters showed higher mCherry expression after induction than the original promoter after induction, with the highest MFI exceeding the original promoter to 857% (Fig. 4c). Please note that we also observed some leakage (it refers to the phenomenon where some optimized PL promoters, which are inducible promoters, exhibit higher expression levels before induction compared to the original promoter before induction) in the mutated promoters before induction (see Supplementary Materials). This is a very normal occurrence because our model's optimization process only considers the final expression levels after induction, and the pre-induction expression levels were not accounted for in the model design or training process.

In conclusion, our biological experiments confirmed the effectiveness and reliability of our model, which can significantly enhance the expression strength of prokaryotic promoters.

The experimental results demonstrate that an important application of our model is the *in silico* directed evolution of promoters. To facilitate biologists' free use of our model, we have built a website, as shown in Fig. 5. The user only needs to input some key parameters such as the promoter sequence, the desired mutation sites and frequency, to utilize the model for promoter optimization.

Discussions

From our perspective, CAPE can potentially play a pivotal role in several areas, including but not limited to the following:

(1) *In silico* directed evolution and library construction of promoters: Our model accurately assesses promoter strength and can precisely identify minor differences in promoter sequences. Hence, we can conduct *in silico* directed evolution of promoters to obtain the desired sequences, facilitating our model's crucial role in promoter optimization. Our model's proficiency in directed evolution has been confirmed through multiple experiments. Additionally, should datasets containing mutated promoter strengths exist, integrating them with our model can refine and tailor more precise models, facilitating the development of experimental promoter libraries.

(2) Enhancing the effectiveness of promoter generation models: Some research focuses on promoter generation models to produce synthetic promoters with improved functionality [21, 28, 46]. These models always face a selection step requiring precise prediction of promoter strength, where our model is likely to enhance the effectiveness of existing promoter generation models.

(3) Other promoter strength-related issues: Numerous issues are related to prokaryotic promoters. For instance, many drugs and metabolites are produced using model organisms like *E. coli* [47, 48]. If our model is employed to direct the evolution of promoters, enhancing their strength, it could potentially reduce the manufacturing costs of related drugs and metabolites in the future.

Regarding the applicability, our CAPE model can be used not only for *E. coli* but also for other prokaryotes. This expansion can be achieved mainly on two levels. First, it has been experimentally confirmed that many promoters from other prokaryotes can function in *E. coli*, such as the PnisA promoter we tested. Therefore, it is possible to directly use our trained model to optimize promoters from other prokaryotes and then introduce them into *E. coli* to function (because our model predicts the relative strength of promoters in *E. coli* to some extent). Second, if promoter sequences and their corresponding strength data from other prokaryotes are

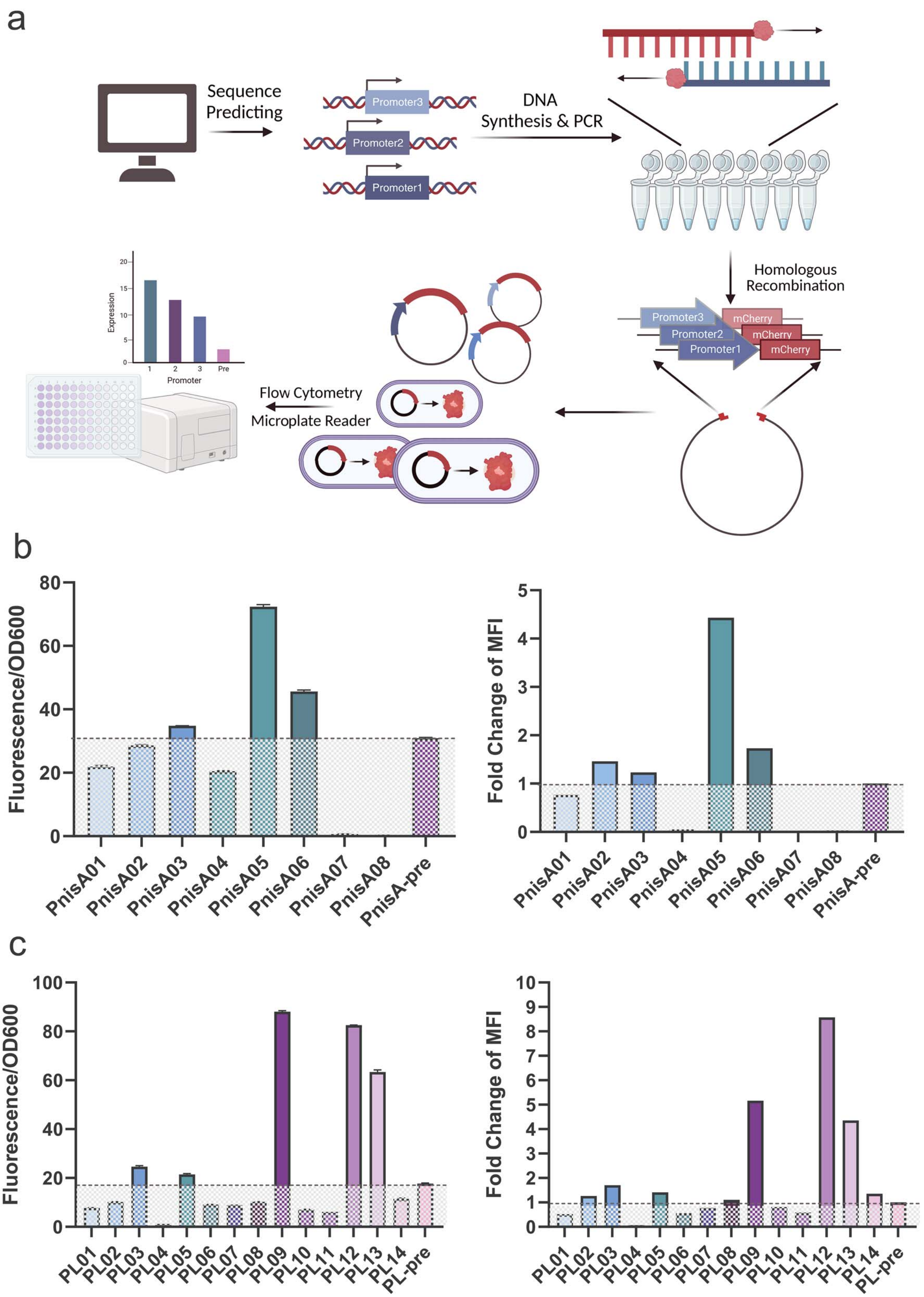
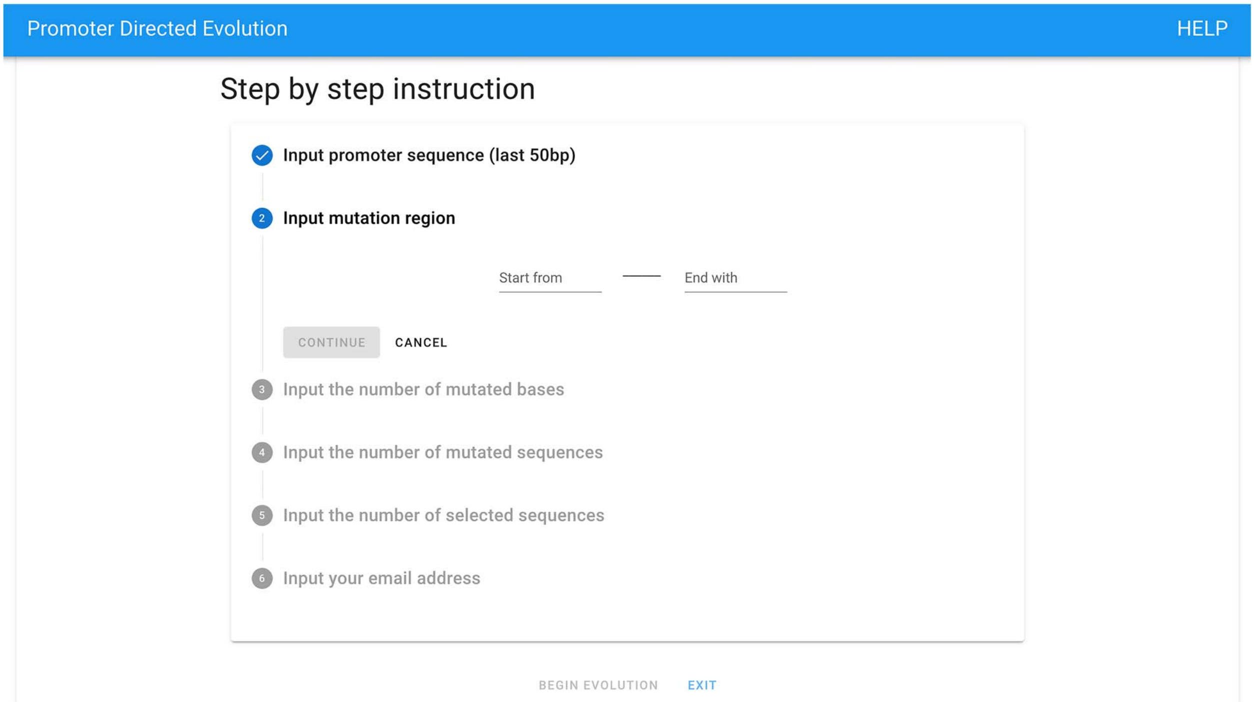


Figure 4. Experimental results. (a) Flowchart of experimental process. (b) Results of PnisA promoter. (c) Results of PL Promoter. Please note that for the data measured by the microplate reader, we had two sets of replicates for each experiment. In the data measured by flow cytometry, MFI refers to the Median Fluorescence Intensity.



Step by step instruction

1. Input promoter sequence (last 50bp)
2. Input mutation region

Start from — End with

3. Input the number of mutated bases
4. Input the number of mutated sequences
5. Input the number of selected sequences
6. Input your email address

Figure 5. Website Page. Users can click the 'HELP' button to view the website usage guide. Follow the prompts on the web page and enter the required parameters step by step, clicking the 'CONTINUE' button after each item to move on to the next item. Once all the parameters have been entered and checked, click the 'BEGIN EVOLUTION' button to get the results. More detailed instruction and the address of the website can be found in https://github.com/BobYHY/CAPE.

available, the CAPE model can be retrained based on the sequence and strength data to enable its application in other species.

Of course, there are some other things to be noticed. Currently, our model only accepts 50-bp sequences for prediction. For larger promoters, using sequences closer to the transcription start site as a substitute has been proven effective. The sequence length limitation primarily stems from the dataset constraints. If more diverse datasets are integrated in the future, our model can further improve by overcoming this limitation through Natural Language Processing (NLP) methods such as padding [26].

In addition, our method primarily focuses on predicting the strength of constitutive promoters and inducible promoters after induction. However, we observed some leakage in the mutated promoters before induction (see Supplementary Materials). This phenomenon can be attributed to sequence alterations that reduced the binding affinity of the original repressor protein with the promoter region, as anticipated. Nevertheless, we also identified mutated promoters that exhibited less pronounced leakage but a significant increase in expression levels after induction. This suggests that we may need to consider the specific kinetic dynamics of the inducible promoters in our subsequent work to enhance the model's reliability. Designing a separate dual-task model based on our model structure could likely help address this issue. Furthermore, excessive leakage can potentially allow us to employ *in silico* directed evolution to transform inducible promoters into constitutive promoters, thereby achieving long-term stable gene expression.

In summary, we proposed a powerful and useful tool, CAPE, to predict the strength of prokaryotic promoters, achieving SOTA performance in two distinct tasks. We have also validated the effectiveness and robustness of this tool through successful fluorescence expression assays. This deep learning model comprehensively understands the essence of promoters by leveraging evolutionary information, holding significant importance. By grasping the biological essence of 'evolution', we have advanced breakthroughs in promoter design and optimization. Additionally, we have provided a website for users to freely utilize our tool.

Key Points

- A deep learning model, CAPE, was constructed using a novel feature extraction method based on evolutionary information to predict promoter strength.
- Our model achieves state-of-the-art results on two distinct types of promoter strength prediction tasks.
- Experimental results of fluorescent protein assay confirm the efficacy of CAPE in simulating *in silico* directed evolution of promoters.

Acknowledgments

Sincere thanks are extended to all members and advisors of Tsinghua iGEM 2022 for their dedicated efforts towards the incubation of this project. Additionally, Professor Stephen S.-T. Yau is grateful to the National Center for Theoretical Sciences (NCTS) for providing an excellent research environment while part of this research was done. We are also grateful to Innovation Laboratory for Undergraduates, Center for Laboratory Training in Life Sciences (CLTLS), Tsinghua University.

Supplementary data

Supplementary data is available at *Briefings in Bioinformatics* online.

Conflict of interest: No competing interest is declared.

Funding

This work is supported in part by funds from National Natural Science Foundation of China (NSFC) grant (12171275), Tsinghua University Education Foundation fund (042202008), Tsinghua University Initiative Scientific Research Program, Academic and Scientific Works Competition for Undergraduates, Academic Affairs Office, Tsinghua University, Xuetang Program, Tsinghua University.

Availability of data and materials

The data presented in this study can be downloaded in the public database, and also available in Supplementary Materials. The Supplementary Materials as well as the source code implemented in this study can be found in our GitHub repository https://github.com/BobYHY/CAPE. Considering potential updates, please refer to our GitHub repository for instructions on accessing the website.

Author contributions statement

Conceptualization, R.R., H.Y., and S.Y.; Methodology, R.R., H.Y., and S.Y.; Investigation: R.R., J.T., and S.M.; Software: R.R., H.Y., Y.T., and Z.B.; Data curation, R.R., Z.B., H.Y. and Y.T.; Formal analysis: R.R., H.Y., and J.T.; Validation: R.R., J.T., S.M., H.Y. and Y.T.; Visualization: R.R., H.Y., and J.T.; Writing—original draft preparation, R.R., H.Y., and J.T.; Writing—review and editing, R.R. and S.Y.; Resources: S.Y. and R.R.; Supervision: S.Y.; Project Administration: S.Y.; Funding Acquisition: S.Y. and R.R. All authors have read and agreed to the published version of the manuscript.

References

1. Slusarczyk AL, Lin A, Weiss R. Foundations for the design and implementation of synthetic genetic circuits. *Nat Rev Genet* 2012;**13**:406–20. https://doi.org/10.1038/nrg3227.
2. Nielsen AAK, Segall-Shapiro TH, Voigt CA. Advances in genetic circuit design: novel biochemistries, deep part mining, and precision gene expression. *Curr Opin Chem Biol* 2013;**17**:878–92. https://doi.org/10.1016/j.cbpa.2013.10.003.
3. Patel RN. Biocatalysis for synthesis of pharmaceuticals. *Bioorg Med Chem* 2018;**26**:1252–74. https://doi.org/10.1016/j.bmc.2017.05.023.
4. Nakagawa A, Matsumura E, Koyanagi T. *et al.* Total biosynthesis of opiates by stepwise fermentation using engineered Escherichia coli. *Nat Commun* 2016;**7**:10390. https://doi.org/10.1038/ncomms10390.
5. Kunjapur AM, Tarasova Y, Prather KLJ. Synthesis and accumulation of aromatic aldehydes in an engineered strain of Escherichia coli. *J Am Chem Soc* 2014;**136**:11644–54. https://doi.org/10.1021/ja506664a.
6. Kalscheuer R, Stolting T, Steinbuchel A. Microdiesel: Escherichia coli engineered for fuel production. *Microbiology* 2006;**152**:2529–36. https://doi.org/10.1099/mic.0.29028-0.
7. Ueki T, Walker DJF, Woodard TL. *et al.* An Escherichia coli chassis for production of electrically conductive protein nanowires. *ACS Synth Biol* 2020;**9**:647–54. https://doi.org/10.1021/acssynbio.9b00506.
8. Cazier AP, Blazeck J. Advances in promoter engineering: novel applications and predefined transcriptional control. *Biotechnol J* 2021;**16**:e2100239. https://doi.org/10.1002/biot.202100239.
9. Hammer K, Mijakovic I, Jensen PR. Synthetic promoter libraries–tuning of gene expression. *Trends Biotechnol* 2006;**24**:53–5. https://doi.org/10.1016/j.tibtech.2005.12.003.
10. Blazeck J, Alper HS. Promoter engineering: recent advances in controlling transcription at the most fundamental level. *Biotechnol J* 2013;**8**:46–58. https://doi.org/10.1002/biot.201200120.
11. Guiziou S, Sauveplane V, Chang H-J. *et al.* A part toolbox to tune genetic expression in Bacillus subtilis. *Nucleic Acids Res* 2016;**44**:gkw624–508. https://doi.org/10.1093/nar/gkw624.
12. De Mey, Maertens J, Lequeux GJ. *et al.* Construction and model-based analysis of a promoter library for e. coli: an indispensable tool for metabolic engineering. *BMC Biotechnol* 2007; **7**:1–14.
13. Portela RMC, Vogl T, Kniely C. *et al.* Synthetic core promoters as universal parts for fine-tuning expression in different yeast species. *ACS Synth Biol* 2017;**6**:471–84. https://doi.org/10.1021/acssynbio.6b00178.
14. Alper H, Fischer C, Nevoigt E. *et al.* Tuning genetic control through promoter engineering. *Proc Natl Acad Sci* 2005;**102**:12678–83. https://doi.org/10.1073/pnas.0504604102.
15. Liu B, Yang F, Huang D-S. *et al.* Ipromoter-2l: a two-layer predictor for identifying promoters and their types by multi-window-based pseknc. *Bioinformatics* 2018;**34**:33–40. https://doi.org/10.1093/bioinformatics/btx579.
16. Song K. Recognition of prokaryotic promoters based on a novel variable-window z-curve method. *Nucleic Acids Res* 2012;**40**:963–71. https://doi.org/10.1093/nar/gkr795.
17. Chevez-Guardado R, Peña-Castillo L. Promotech: a general tool for bacterial promoter recognition. *Genome Biol* 2021;**22**:1–16.
18. Xiao X, Zhao-Chun X, Qiu W-R. *et al.* Ipsw (2l)-pseknc: a two-layer predictor for identifying promoters and their strength by hybrid features via pseudo k-tuple nucleotide composition. *Genomics* 2019;**111**:1785–93. https://doi.org/10.1016/j.ygeno.2018.12.001.
19. Tahir M, Hayat M, Gul S. *et al.* An intelligent computational model for prediction of promoters and their strength via natural language processing. *Chemom Intel Lab Syst* 2020;**202**:104034. https://doi.org/10.1016/j.chemolab.2020.104034.
20. Qiao H, Zhang S, Xue T. *et al.* Ipro-Gan: a novel model based on generative adversarial learning for identifying promoters and their strength. *Comput Methods Programs Biomed* 2022;**215**:106625. https://doi.org/10.1016/j.cmpb.2022.106625.
21. Wang Y, Wang H, Wei L. *et al.* Synthetic promoter design in Escherichia coli based on a deep generative network. *Nucleic Acids Res* 2020;**48**:6403–12. https://doi.org/10.1093/nar/gkaa325.
22. Jumper J, Evans R, Pritzel A. *et al.* Highly accurate protein structure prediction with alphafold. *Nature* 2021;**596**:583–9. https://doi.org/10.1038/s41586-021-03819-2.
23. Wang Y, Marrero MC, Medema MH. *et al.* Coevolution-based prediction of protein–protein interactions in polyketide biosynthetic assembly lines. *Bioinformatics* 2020;**36**:4846–53. https://doi.org/10.1093/bioinformatics/btaa595.
24. Hoang T, Yin C, Yau SS-T. Numerical encoding of dna sequences by chaos game representation with application in similarity comparison. *Genomics* 2016;**108**:134–42. https://doi.org/10.1016/j.ygeno.2016.08.002.
25. Huang G, Liu Z, Van Der Maaten L. *et al.* Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA: IEEE, 2017, 2261–69.
26. Vaswani A, Shazeer N, Parmar N. *et al.* Attention is all you need. *Adv Neural Inf Process Syst* 2017;**30**:6000–10.
27. Thomason MK, Bischler T, Eisenbart SK. *et al.* Global transcriptional start site mapping using differential rna

sequencing reveals novel antisense rnas in Escherichia coli. *J Bacteriol* 2015;**197**:18–28. https://doi.org/10.1128/JB.02096-14.

28. Zhang P, Wang H, Hanwen X. *et al*. Deep flanking sequence engineering for efficient promoter design using deepseed. *Nat Commun* 2023;**14**:6309, 1–14. https://doi.org/10.1038/s41467-023-41899-y.
29. Herman A, Boer D, Comstock LJ. *et al*. The tac promoter: a functional hybrid derived from the trp and lac promoters. *Proc Natl Acad Sci* 1983;**80**:21–5.
30. Brosius J, Erfle M, Storella J. Spacing of the-10 and-35 regions in the tac promoter. Effect on its in vivo activity. *J Biol Chem* 1985; **260**:3539–41. https://doi.org/10.1016/S0021-9258(19)83655-4.
31. Zhao M, Yuan Z, Longtao W. *et al*. Precise prediction of promoter strength based on a de novo synthetic promoter library coupled with machine learning. *ACS Synth Biol* 2021;**11**:92–102. https://doi.org/10.1021/acssynbio.1c00117.
32. Yang W, Li D, Huang R. Evmp: enhancing machine learning models for synthetic promoter strength prediction by extended vision mutant priority framework. *Front Microbiol* 2023;**14**:07. https://doi.org/10.3389/fmicb.2023.1215609.
33. Wei S, Liu M-L, Yang Y-H. *et al*. Ppd: a manually curated database for experimentally verified prokaryotic promoters. *J Mol Biol* 2021;**433**:166860.
34. Altschul S, Gish W, Miller W. *et al*. Basic local aligment search tool. *J Mol Biol* 1990;**215**:403–10. https://doi.org/10.1016/S0022-2836(05)80360-2.
35. Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol* 1970;**48**:443–53. https://doi.org/10.1016/0022-2836(70)90057-4.
36. Ren R, Yin C, Yau SS-T. kmer2vec: a novel method for comparing dna sequences by word2vec embedding. *J Comput Biol* 2022;**29**: 1001–21. https://doi.org/10.1089/cmb.2021.0536.
37. Blaisdell BE. A measure of the similarity of sets of sequences not requiring sequence alignment. *Proc Natl Acad Sci* 1986;**83**:5155–9. https://doi.org/10.1073/pnas.83.14.5155.
38. Tiee-Jian W, Burke JP, Davison DB. A measure of dna sequence dissimilarity based on mahalanobis distance between frequencies of words. *Biometrics* 1997;**53**(4):1431–9.
39. Kantorovitz MR, Robinson GE, Sinha S. A statistical method for alignment-free comparison of regulatory sequences. *Bioinformatics* 2007;**23**:i249–55. https://doi.org/10.1093/bioinformatics/btm211.
40. Mikolov T, Chen K, Corrado G. *et al*. Efficient estimation of word representations in vector space arXiv preprint arXiv:1301.3781. 2013. https://doi.org/10.48550/arXiv.1301.3781.
41. Mikolov T, Sutskever I, Chen K. *et al*. Distributed representations of words and phrases and their compositionality. *Adv Neural Inf Process Syst* 2013;**2**:3111–9.
42. He K, Zhang X, Ren S. *et al*. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, 2016, 770–8.
43. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Commun ACM* 2012;**60**: 84–90.
44. Elman JL. Finding structure in time. *Cognit Sci* 1990;**14**:179–211. https://doi.org/10.1207/s15516709cog1402_1.
45. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* 1997;**9**:1735 12–80. https://doi.org/10.1162/neco.1997.9.8.1735.
46. Seo E, Choi Y-N, Shin YR. *et al*. Design of synthetic promoters for cyanobacteria with generative deep-learning model. *Nucleic Acids Res* 2023;**51**:7071–82. https://doi.org/10.1093/nar/gkad451.
47. Baeshen MN, Al-Hejin AM, Bora RS. *et al*. Production of biopharmaceuticals in e. coli: current scenario and future perspectives. *J Microbiol Biotechnol* 2015;**25**:953–62. https://doi.org/10.4014/jmb.1412.12079.
48. Yang D, Park SY, Park YS. *et al*. Metabolic engineering of Escherichia coli for natural product biosynthesis. *Trends Biotechnol* 2020;**38**:745–65. https://doi.org/10.1016/j.tibtech.2019.11.007.