Mathematical Biosciences and Engineering

MBE, 19(12): 13294–13305.
DOI: 10.3934/mbe.2022622
Received: 04 July 2022
Revised: 23 August 2022
Accepted: 29 August 2022
Published: 13 September 2022

http://www.aimspress.com/journal/MBE

Research article

SPREAD: An ensemble predictor based on DNA autoencoder framework for discriminating promoters in *Pseudomonas aeruginosa*

Shengming Zhou, Jia Zheng* and Cangzhi Jia*

School of Science, Dalian Maritime University, Dalian 116026, China

***Correspondence:** Email: zhengjia@dlmu.edu.cn, cangzhijia@dlmu.edu.cn.

Abstract: Regulatory elements in DNA sequences, such as promoters, enhancers, terminators and so on, are essential for gene expression in physiological and pathological processes. A promoter is the specific DNA sequence that is located upstream of the coding gene and acts as the "switch" for gene transcriptional regulation. Lots of promoter predictors have been developed for different bacterial species, but only a few are designed for *Pseudomonas aeruginosa*, a widespread Gram-negative conditional pathogen in nature. In this work, an ensemble model named SPREAD is proposed for the recognition of promoters in *Pseudomonas aeruginosa*. In SPREAD, the DNA sequence autoencoder model LSTM is employed to extract potential sequence information, and the mean output probability value of CNN and RF is applied as the final prediction. Compared with G4PromFinder, the only state-of-the-art classifier for promoters in *Pseudomonas aeruginosa*, SPREAD improves the prediction performance significantly, with an accuracy of 0.98, recall of 0.98, precision of 0.98, specificity of 0.97 and F1-score of 0.98.

Keywords: promoters; autoencoder model; machine learning; deep learning

1. Introduction

Pseudomonas aeruginosa (P. aeruginosa), a kind of Gram-negative bacteria, is an opportunistic pathogen able to cause serious infections and has become the third source of nosocomial infection [1,2]. In addition, the medical treatment for such infections is fairly difficult due to the high level of innate and acquired antibiotic resistance of *Pseudomonas aeruginosa* [3,4]. The pathogenesis of *Pseudomonas aeruginosa* is modulated by signaling pathways [5,6]. A certain adaptation mechanism

is necessary for *Pseudomonas aeruginosa* to enter the body and survive, which involves regulatory elements such as promoters, enhancers, terminators and so on [7]. Among these regulatory elements, the promoter, the special DNA sequence located upstream of the coding gene, can be recognized and combined with a special RNA polymerase, and as a result it plays the role of "switch" for gene transcriptional regulation [8,9]. Therefore, identifying promoter fragments accurately is basic and essential for the further study of gene regulation in *Pseudomonas aeruginosa*.

With the availability of high-throughput screening data, many computational methods have been developed to identify promoters in bacterial genomes, such as *Escherichia coli*. [10–24]. On one hand, both shallow machine learning and deep learning classifiers are used to build promoter recognition models. More specifically, BPROM [19] applied linear discriminant analysis (LDA). 70ProPred [14], iPromoter-2L [17], IBPP [22], iProEP [15] and MULTiPLy [23] all applied a support vector machine (SVM) as the classifier. Promotech [11] used a random forest (RF) and recurrent neural network (RNN) to predict bacterial promoters. In Stack-ORI [16] 12 schemes of feature coding were used to train the eXtreme Gradient Boosting (XGBoost) algorithm, and the final model was built based on the stacked ensemble approach. CNNProm [21] and iPromoter-BnCNN [10] adopted convolutional neural networks (CNN). On the other hand, information of different conservative motifs or certain physicochemical characteristics of DNA sequences are employed as feature representation in bacterial promoter detection algorithms. BPROM [19] applied five relatively conserved sequence motifs and two sequence features. G4PromFinder [13], which focused on the promoter recognition of *Streptomyces coelicolor* A3(2) and *Pseudomonas aeruginosa* PA14, took advantage of AT-rich elements and G-quadruplex motifs, these two conserved sequence features in the DNA sequence, to build a prediction algorithm. G4PromFinder outperforms three other promoter prediction algorithms, bTSSfinder [20], PePPER [12] and PromPredict [18], in the bacterial genome of GC-rich. In particular, most of these fourteen models are built for *Escherichia coli*, and G4PromFinder is the only one available to recognize promoters in *Pseudomonas aeruginosa*. However, the prediction results of G4PromFinder just achieved a recall of 69.0%, a precision of 43.1%, a specificity of 8.9%, an F1-score of 53% and an accuracy of 38.9%, so there is ample room for improvement.

In this study, a novel model, SPREAD, is introduced to identify promoters in *Pseudomonas aeruginosa*. Specifically, for feature extraction, the DNA sequence autoencoder model LSTM and an improved teacher forcing mechanism are trained based on the promoter and non-promoter sequence data of *Pseudomonas aeruginosa* PA14, in order to mine the potential information of DNA sequences. Their outputs form embedding vectors and matrices in different sizes. Embedding vectors are inputted into five traditional machine learning algorithms, containing XGBoost, RF, GNB, SVM, KNN, respectively, and embedding matrices are inputted to a deep learning CNN architecture. Different embedding dimensions and different integrations of classifiers are evaluated. After comparison and optimization, the combination of RF with an embedding vector of dimension 256 and CNN with embedding matrix of dimensions 81*32 shows the best prediction performance, and forms the final ensemble prediction model. Extensive results show that SPREAD is superior to other existing methods and can be a powerful auxiliary tool to screen *Pseudomonas aeruginosa* promoters in batches for experimenters.

2. Materials and methods

2.1. Datasets processing

The genomic sequences of *P. aeruginosa* were collected from the National Center for Biotechnology Information (NCBI) under accession code NC_008463.1. Then, according to the annotation information together with transcriptome information provided by Wurtzel O. et al. [25], 2117 sequences with the TSS site at the 0-th position were extracted 60 bp upstream and 20 bp downstream. In order to decrease redundancy, CD-HIT-EST [26] was applied to prune those sequences with homology above 0.8. Finally, there were 2108 sequences retained for our positive dataset. To build the negative dataset, 10,000 sequences of length 81 without TSS sites were randomly selected in the same genome. Then, we deleted the sequences whose similarities with positive samples were greater than 0.8. Finally, 2108 negative sequences were chosen randomly to build the negative dataset. The samples are available in the Supplementary file.

2.2. DNA autoencoder: LSTM

Traditional sequence features have to be defined manually in advance based on experience, before being derived from the sequence itself or its physicochemical properties [27]. Therefore, it is sometimes impossible to fully mine the potential information from the sequences, which may affect the performance of the prediction model [28]. Different from traditional methods, neural networks with many variants can automatically capture features from sequences and make a classification accordingly [29–31]. Through unsupervised learning, an autoencoder composed of neural networks is able to mine potential information out of input data and represent it efficiently [32,33]. This efficient representation of input data is called coding, and its dimension is usually much smaller than that of input data, which means the autoencoder is helpful for dimensionality reduction [34]. Features extracted by the autoencoder can be used alone directly or combined with traditional manual features, so as to improve the performance of the predictor.

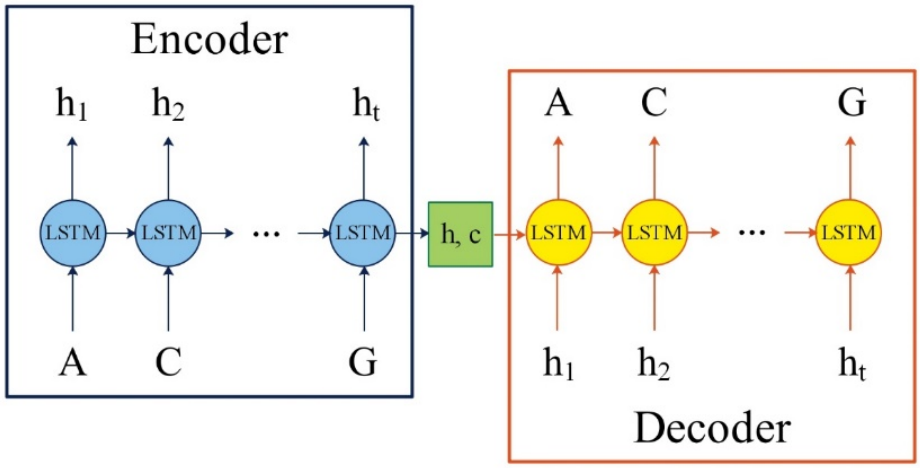

Figure 1. DNA autoencoder.

Considering the above advantages of an autoencoder, the DNA autoencoder model LSTM was utilized in this work to encode the efficient representation of input data through unsupervised learning [32]. The DNA automatic encoder framework we constructed was based on a sequence to sequence (seq2seq) structure [35]. The encoder and decoder both consisted of long short-term memory networks (LSTM) and a modified teacher forcing mechanism, as shown in Figure 1. As a modification of the traditional teacher forcing mechanism, inputs of decoder nodes were the outputs of the LSTM encoder nodes, instead of the prediction result of the previous LSTM node in the decoder. The output of this autoencoder is embedding matrices and vectors.

2.3. Machine learning and deep learning algorithms

In this work, five traditional machine learning models, including eXtreme Gradient Boosting (XGBoost) [36], random forest (RF) [37], Gaussian Naive Bayes (GNB) [38], support vector machine (SVM) [39] and k nearest neighbors (KNN) [40], combined with a deep learning algorithm, i.e., a convolutional neural network (CNN) [41], were tried to classify promoters and non-promoters in *P. aeruginosa*.

In order to obtain the optimal performance, we adopted the method of adjusting parameters. For CNN, three parameters, "filters," "conv_size" and "latent_dim," were selected in sets [64, 128, 256, 512], [3, 5, 7], [128, 256, 512], respectively. For XGBoost, RF and KNN, the parameter "n_estimators" was selected in sets [100, 200, 300, 400, 500], [100, 200, 300, 400, 500] and [3, 4, 5, 6, 7, 8, 9, 10, 11, 12], respectively. For SVM, we only focused on the parameter "kernel," selected in ["linear," "poly," "rbf"]. The final parameters used in each model are available in Tables S1–S5.

2.4. Performance evaluation

Evaluation indicators, including recall (sensitivity), specificity, precision, F1-score and accuracy, were adopted to assess and compare the performance of our model on the training and validation datasets. The formulas are calculated as below:

$$\begin{cases} Recall = \dfrac{TP}{FN + TP} \\ Specificity = \dfrac{TN}{FP + TN} \\ Precision = \dfrac{TP}{FP + TP} \\ F1\text{-}score = \dfrac{2 \times \mathrm{Re}call \times \mathrm{Pr}ecision}{\mathrm{Re}call + \mathrm{Pr}ecision} \\ Accuracy = \dfrac{TP + TN}{TP + TN + FN + FP} \end{cases} \tag{1}$$

where TP, TN, FP and FN represent the numbers of true positives, true negatives, false positives and false negatives, respectively. In addition, the area under the receiver operating characteristic curve (AUC) is also used to examine the performance of our model. Among the seven measurement indicators, the recall, precision and F1-score can well describe the model's ability to predict true promoters, and the accuracy and AUC can reflect the overall performance of the model.

3. Results and discussion

3.1. Optimization of DNA autoencoder: LSTM

Because of the superiority of neural network architectures, the autoencoder technique composed of an encoder and decoder was proposed as an unsupervised learning tool to capture the potential information and obtain the low-dimensional vector representation of the DNA sequences [34]. During the optimization of the automatic encoder, the sizes of output embedding vectors and matrices are key elements that influence the classification performance. For one output embedding vector (E_{vec}), we set the embedding size as 16, 32, 64, 128, 256, 512 and 1024, based on experience. For the output embedding matrix (E_{mat}) with $L \times W$ dimensions, L is the fixed sequence length 81, and W represents the dimension of the embedding feature, which was set as 8, 16, 32.

3.2. Establishment of prediction model

In order to find the most suitable model for identifying promoters of *P. aeruginosa*, we tested and compared each embedding size and classification algorithm using five evaluation indicators (Recall, Precision, F1-score, Accuracy, AUC) and a 10-fold cross-validation test. Specifically, embedding vectors were inputted into five traditional machine learning algorithms, containing XGBoost, RF, GNB, SVM, KNN, respectively, and embedding matrices were inputted to a deep learning CNN architecture.

We choose the appropriate size according to the performance of embedding vectors and matrices of different sizes on the six models. For embedding vectors and matrices, the models' performances are shown in Table 1. It is coincident that these four traditional models (XGBoost, GNB, SVM, KNN) all achieve their best performance when the dimension of vector Evec is 512, except RF achieves its best performance when the dimension of vector Evec is 256. The CNN model reaches its best performance for the matrix Emat with dimensions 81*32. Furthermore, the AUC values of the six best models were compared on 10-fold cross-validation, and the CNN model based on an embedding matrix with dimensions 81*32 achieves the best AUC of 0.95(±0.01), which is 0.05 higher than those of the second-place SVM models. Figure 2 demonstrates the performances of the six models according to AUC value.

Table 1. Five evaluation indicators of the five traditional machine learning algorithms (XGBoost, RF, GNB, SVM, KNN) and one deep learning network CNN.

Model	Best size	Recall	Precision	F1-score	Accuracy	AUC
XGBoost	512	0.86(±0.02)	0.82(±0.01)	0.84(±0.02)	0.84(±0.01)	0.89(±0.02)
RF	256	0.85(±0.03)	0.83(±0.01)	0.84(±0.02)	0.84(±0.02)	0.89(±0.01)
GNB	512	0.73(±0.03)	0.81(±0.02)	0.77(±0.02)	0.78(±0.02)	0.86(±0.01)
SVM	512	0.87(±0.03)	0.84(±0.02)	0.85(±0.02)	0.85(±0.02)	0.90(±0.01)
KNN	512	0.89(±0.02)	0.83(±0.02)	0.86(±0.02)	0.85(±0.02)	0.88(±0.01)
CNN	81*32	0.87(±0.03)	0.89(±0.02)	0.88(±0.01)	0.88(±0.01)	0.95(±0.01)

The integration of the above six classifiers was carried out in order to further optimize the prediction model. With each attempt, the mean output probabilities were calculated as the final predicted value. With thresholds set at 0.5, 0.55 and 0.6, respectively, if the mean value is greater than

the threshold, the sample will be considered as a promoter; otherwise, the sample will be considered as a non-promoter.

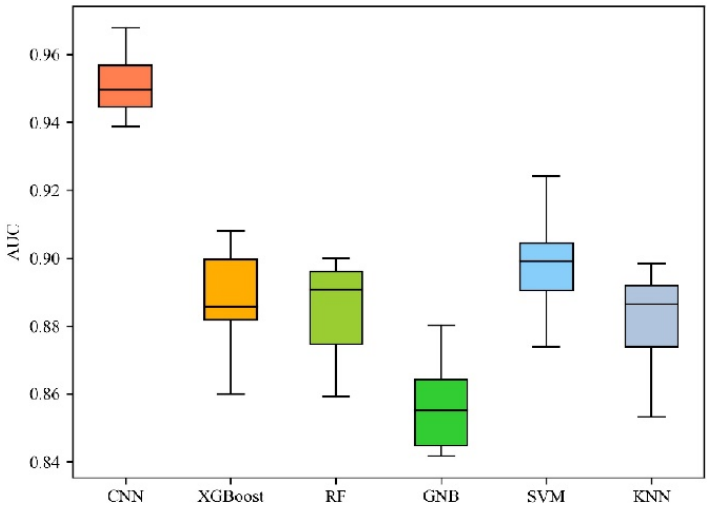

Figure 2. The performance comparison of six models based on the AUC values.

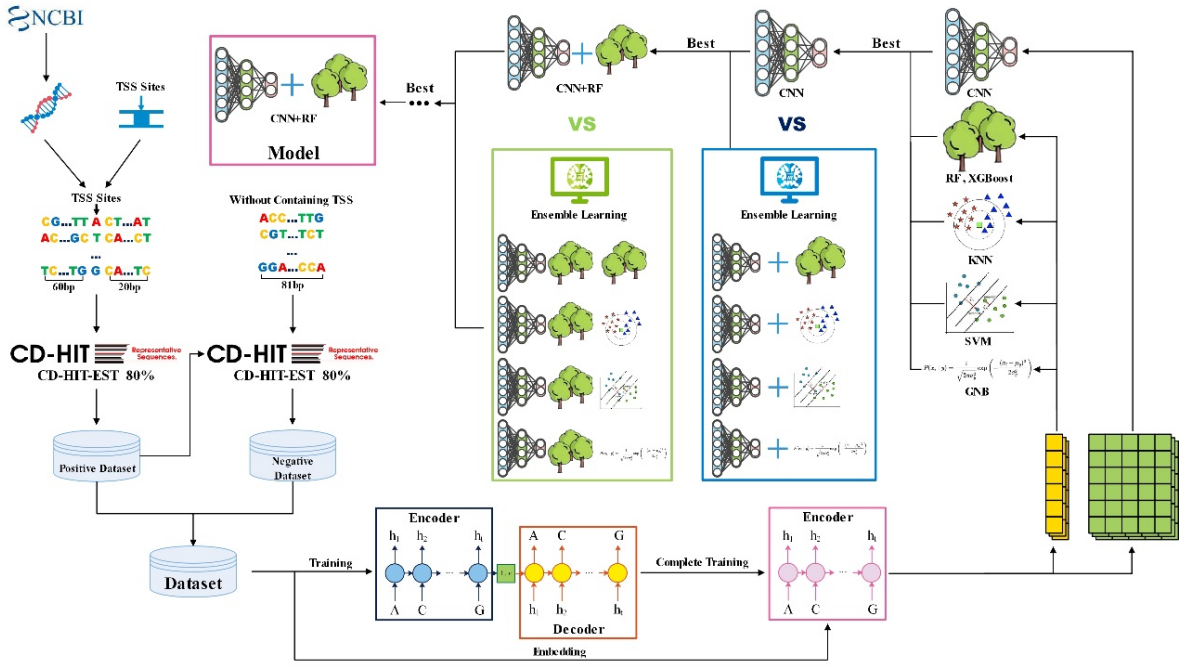

Figure 3. The workflow of the promoter predictor for *P. aeruginosa*.

First, we integrated the CNN with the other five models and compared their prediction results on a 10-fold cross-validation test (Table S6). It can be noticed that the integration of CNN and RF on the threshold of 0.5 reaches the best precision of 0.88, accuracy of 0.89 and AUC of 0.95. The combination of CNN and SVM, CNN and KNN are simultaneously achieved the second best precision of 0.87, accuracy of 0.89 and AUC of 0.95. Then, we added the other four classifiers to CNN and RF integration, successively (Table S6). However, the performance of these integrated prediction models

is not improved at all. We have tried the stacking strategy of fourteen different combinations using 10-fold cross-validation and listed the corresponding prediction results in Table S6. It can be seen that the combination of CNN, RF and SVM reached the best performance with the average AUC of 0.89. However, it is still lower than the integration of CNN and RF on the threshold of 0.5. In addition, we also tried the stacking strategy of fourteen different combinations using 10-fold cross-validation and listed the corresponding prediction results in Table S6. It can be seen that the combination of CNN, RF and SVM reached the best performance with the average AUC of 0.89. However, it is still lower than our model SPREAD with AUC of 0.95. Therefore, our final predictor is designed as the integration of the CNN and RF with the threshold of 0.5. The workflow of this promoter predictor for *P. aeruginosa* is shown in Figure 3.

3.3. Further evaluation of the predictor

In this subsection, we further evaluate our model by comparing it with the latest model G4PromFinder [13], which is superior to PePPER, PromPredict and bTSSfinder according to the reports in [12,18,20]. The same evaluation strategy was repeated 10 times. At each time, 338 positive samples and 338 negative samples were randomly selected as a validation dataset, and the other samples were used as the training dataset. Evaluation indicators of 10 validation datasets were calculated, and for each indicator, the average value is shown in Figure 4. It can be concluded that our model is rather stable and shows better performances on all of five indicators. Specifically, our model reaches an Accuracy of 0.98, a recall of 0.98, a precision of 0.98 and an F1-score of 0.98, which are about 0.59, 0.29, 0.55 and 0.45 higher than those of G4PromFinder, respectively.

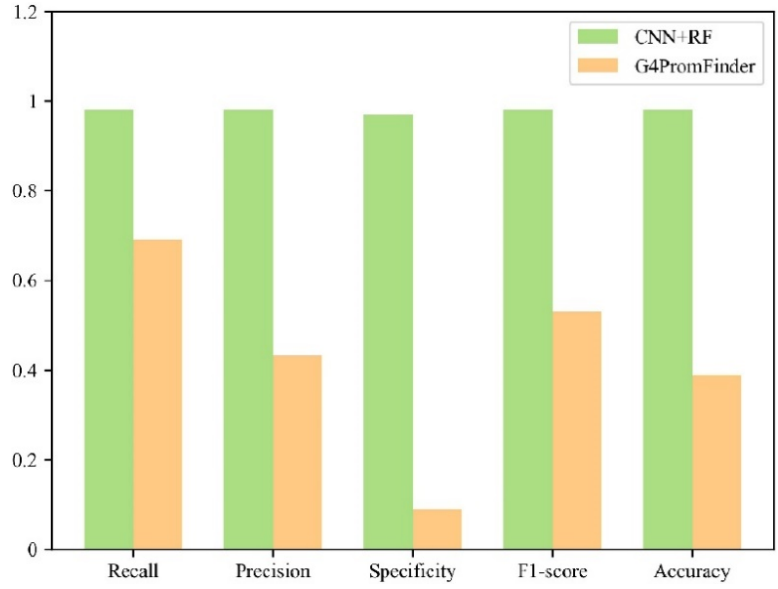

Figure 4. Comparison of our model with G4PromFinder.

In addition, six single classifiers, i.e., XGBoost, RF, GNB, SVM, KNN and CNN, combined with the autoencoder were evaluated and compared with G4PromFinder, in order to assess the effectiveness of the DNA encoder. It is demonstrated in Figure 5 that all of the six autoencoder-based single classifiers are superior to G4PromFinder significantly on all five evaluation indicators, with accuracies

all above 0.77. As for the other indicators, the comparison results are listed in Table S7. These results illustrate that the DNA autoencoder plays a key role in the construction of the efficient predictor.

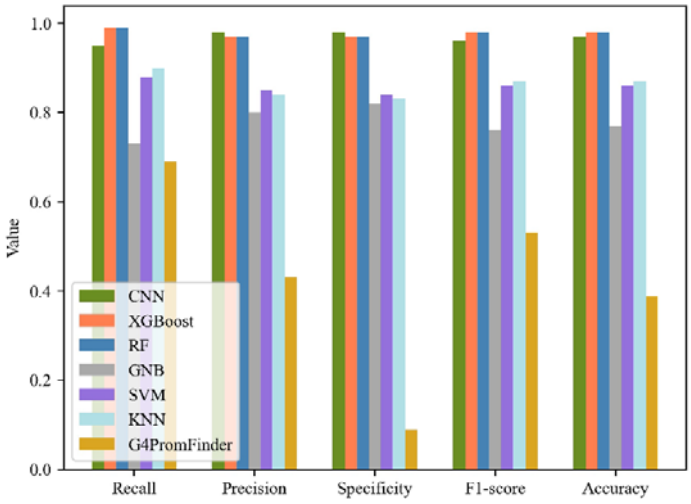

Figure 5. The five indicators of six autoencoder-based individual classifiers and G4PromFinder.

4. Generalization ability of the model

In this section, we applied the same encoding scheme and classification algorithms to distinguish promoters of *Escherichia coli (E. coli)*. The training and independent test datasets are obtained from the latest predictor PredPromoter-MF(2L) [42]. The results of six individual classification algorithms on the training dataset are listed in Table S8. It can be seen that the model based on the CNN reached the best performance with an Accuracy of 0.83 and an AUC of 0.93. Thus, it is used to test different types of promoters and compared with PredPromoter-MF(2L). As illustrated in Table 2, SPREAD is superior to PredPromoter-MF (2L) in prediction of σ^{24}, σ^{28}, σ^{32}, σ^{38}, but it is lower performing in prediction of σ^{54}, σ^{70}. The outperformance indicated the generalization ability of the model.

Table 2. Results comparison of SPREAD and PredPromoter-MF(2L) on specific types of promoters in *E. coli*.

Promoter	Method	TP	FN
σ^{24}	PredPromoter-MF	15	12
	CNN	22	5
σ^{28}	PredPromoter-MF	4	5
	CNN	8	1
σ^{32}	PredPromoter-MF	12	9
	CNN	21	0
σ^{38}	PredPromoter-MF	84	69
	CNN	145	8
σ^{54}	PredPromoter-MF	4	2
	CNN	3	3
σ^{70}	PredPromoter-MF	329	8
	CNN	312	25

5. Conclusions

In this study, we proposed a specific prediction model for promoters in *P. aeruginosa*, called SPREAD, which is an ensemble model based on the integration of a CNN and RF. The effectiveness of our prediction model SPREAD is verified through several different tests, and it improves the accuracy by 0.59 compared to G4PromFinder. SPREAD utilized an LSTM induced matrix and vector embedding of DNA sequences as feature encoding. The test results suggest that the DNA sequence autoencoder strategy contributes to the impressive classification performance. The code and datasets are available at https://github.com/Zhou-Shengming/SPREAD. In the future, this method will be extended to other types of bacterial promoter recognition.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities 3132022204.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. N. Masuda, E. Sakagawa, S. Ohya, Outer membrane proteins responsible for multiple drug resistance in *Pseudomonas aeruginosa*, *Antimicrob. Agents Chemother.*, **39** (1995), 645–649. https://doi.org/10.1128/AAC.39.3.645
2. K. Poole, Multidrug efflux pumps and antimicrobial resistance in *Pseudomonas aeruginosa* and related organisms, *J. Mol. Microbiol. Biotechnol.*, **3** (2001), 255–264.
3. G. Bonfiglio, Y. Laksai, L. Franchino, G. Amicosante, G. Nicoletti, Mechanisms of beta-lactam resistance amongst *Pseudomonas aeruginosa* isolated in an Italian survey, *J. Antimicrob. Chemother*., **42** (1998), 697–702. https://doi.org/10.1093/jac/42.6.697
4. K. Ohlsen, W. Ziebuhr, K. P. Koller, W. Hell, T. A. Wichelhaus, J. Hacker, Effects of subinhibitory concentrations of antibiotics on alpha-toxin (hla) gene expression of methicillin-sensitive and methicillin-resistant Staphylococcus aureus isolates, *Antimicrob. Agents Chemother.*, **42** (1998), 2817–2823. https://doi.org/10.1128/AAC.42.11.2817
5. N. Bagge, O. Ciofu, M. Hentzer, J. I. A. Campbell, M. Givskov, N. Hoiby, Constitutive high expression of chromosomal β-lactamase in *Pseudomonas aeruginosa* caused by a new insertion sequence (IS 1669) located in ampD, *Antimicrob. Agents Chemother.*, **46** (2002), 3406–3411. https://doi.org/10.1128/AAC.46.11.3406-3411.2002
6. P. M. Lepper, E. Grusa, H. Reichl, J. Hogel, M. Trautmann, Consumption of imipenem correlates with β-lactam resistance in *Pseudomonas aeruginosa*, *Antimicrob. Agents Chemother.*, **46** (2002), 2920–2925. https://doi.org/10.1128/AAC.46.9.2920-2925.2002
7. K. J. Hampel, A. E. LaBauve, J. A. Meadows, L. F. Fitzsimmons, A. M. Nock, M. J. Wargo, Characterization of the GbdR regulon in *Pseudomonas aeruginosa*, *J. Bacteriol.*, **196** (2014), 7–15. https://doi.org/10.1128/JB.01055-13

8. L. A. Gallarato, D. G. Sanchez, L. Olvera, E. D. Primo, M. N. Garrido, P. R. Beassoni, et al., Exopolyphosphatase of *Pseudomonas aeruginosa* is essential for the production of virulence factors, and its expression is controlled by NtrC and PhoB acting at two interspaced promoters, *Microbiology-(UK)*, **160** (2014), 406–417. https://doi.org/10.1099/mic.0.074773-0
9. W. Liu, Y. Jiang, H. R. Tang, Inferring gene regulatory networks using the improved Markov blanket discovery algorithm, *Interdiscip. Sci.*, **14** (2022), 168–181. https://doi.org/10.1007/s12539-021-00478-9
10. R. Amin, C. R. Rahman, S. Ahmed, M. H. R. Sifat, M. N. K. Liton, M. M. Rahman, et al., iPromoter-BnCNN: A novel branched CNN-based predictor for identifying and classifying sigma promoters, *Bioinformatics*, **36** (2020), 4869–4875. https://doi.org/10.1093/bioinformatics/btaa609
11. R. Chevez-Guardado, L. Peña-Castillo, Promotech: A general tool for bacterial promoter recognition, *Genome Biol.*, **22** (2021), 1–16. https://doi.org/10.1186/s13059-021-02514-9
12. A. de Jong, H. Pietersma, M. Cordes, O. P. Kuipers, J. Kok, PePPER: A webserver for prediction of prokaryote promoter elements and regulons, *BMC Genomics*, **13** (2012), 1–10. https://doi.org/10.1186/1471-2164-13-299
13. M. D. Salvo, E. Pinatel, A. Tala, M. Fondi, C. Peano, P. Alifano, G4PromFinder: An algorithm for predicting transcription promoters in GC-rich bacterial genomes based on AT-rich elements and G-quadruplex motifs, *BMC Bioinf.*, **19** (2018), 1–11. https://doi.org/10.1186/s12859-018-2049-x
14. W. Y. He, C. Z. Jia, Y. C. Duan, Q. Zou, 70ProPred: A predictor for discovering sigma70 promoters based on combining multiple features, *BMC Syst. Biol.*, **12** (2018), 99–107. https://doi.org/10.1186/s12918-018-0570-1
15. H. Y. Lai, Z. Y. Zhang, Z. D. Su, W. Su, H. Ding, W. Chen, et al., iProEP: A computational predictor for predicting promoter, *Mol. Ther. Nucleic Acids*, **17** (2019), 337–346. https://doi.org/10.1016/j.omtn.2019.05.028
16. F. Y. Li, J. X. Chen, Z. Y. Ge, Y. Wen, Y. W. Yue, M. Hayashida, et al., Computational prediction and interpretation of both general and specific types of promoters in Escherichia coli by exploiting a stacked ensemble-learning framework, *Briefings Bioinf.*, **22** (2021), 2126–2140. https://doi.org/10.1093/bib/bbaa049
17. B. Liu, F. Yang, D. S. Huang, K. C. Chou, iPromoter-2L: A two-layer predictor for identifying promoters and their types by multi-window-based PseKNC, *Bioinformatics*, **34** (2018), 33–40. https://doi.org/10.1093/bioinformatics/btx579
18. V. Rangannan, M. Bansal, High-quality annotation of promoter regions for 913 bacterial genomes, *Bioinformatics*, **26** (2010), 3043–3050. https://doi.org/10.1093/bioinformatics/btq577
19. V. Salamov, A. Solovyevand, Automatic annotation of microbial genomes and metagenomic sequences, in *Metagenomics and its Applications in Agriculture, Biomedicine and Environmental Studies*, (2011), 61–78.
20. I. A. Shahmuradov, R. M. Razali, S. Bougouffa, A. Radovanovic, V. B. Bajic, bTSSfinder: A novel tool for the prediction of promoters in cyanobacteria and Escherichia coli, *Bioinformatics*, **33** (2017), 334–340. https://doi.org/10.1093/bioinformatics/btw629
21. R. K. Umarov, V. V. Solovyev, Recognition of prokaryotic and eukaryotic promoters using convolutional deep learning neural networks, *PLoS One*, **12** (2017), e0171410. https://doi.org/10.1371/journal.pone.0171410

22. S. Wang, X. S. Cheng, Y. J. Li, M. Wu, Y. H. Zhao, Image-based promoter prediction: A promoter prediction method based on evolutionarily generated patterns, *Sci. Rep.*, **8** (2018), 1–9. https://doi.org/10.1038/s41598-018-36308-0
23. M. Zhang, F. Y. Li, T. T. Marquez-Lago, A. Leier, C. Fan, C. K. Kwoh, et al., MULTiPly: A novel multi-layer predictor for discovering general and specific types of promoters, *Bioinformatics*, **35** (2019), 2957–2965. https://doi.org/10.1093/bioinformatics/btz016
24. W. Su, M. L. Liu, Y. H. Yang, J. S. Wang, S. H. Li, H. Lv, et al., PPD: A manually curated database for experimentally verified prokaryotic promoters, *J. Mol. Biol.*, **433** (2021), 166860. https://doi.org/10.1016/j.jmb.2021.166860
25. O. Wurtzel, D. R. Yoder-Himes, K. Han, A. A. Dandekar, S. Edelheit, E. P. Greenberg, et al., The single-nucleotide resolution transcriptome of *Pseudomonas aeruginosa* grown in body temperature, *PLoS Pathog.*, **9** (2012), e1002945. https://doi.org/10.1371/journal.ppat.1002945
26. Y. Huang, B. F. Niu, Y. Gao, L. M. Fu, W. Z. Li, CD-HIT Suite: A web server for clustering and comparing biological sequences, *Bioinformatics*, **26** (2010), 680–682. https://doi.org/10.1093/bioinformatics/btq003
27. R. P. Xie, J. H. Li, J. W. Wang, W. Dai, A. Leier, T. T. Marquez-Lago, et al., DeepVF: A deep learning-based hybrid framework for identifying virulence factors using the stacking strategy, *Briefings Bioinf.*, **22** (2021), bbaa125. https://doi.org/10.1093/bib/bbaa125
28. D. D. Zheng, G. S. Pang, B. Liu, L. H. Chen, J. Yang, Learning transferable deep convolutional neural networks for the classification of bacterial virulence factors, *Bioinformatics*, **36** (2020), 3693–3702. https://doi.org/10.1093/bioinformatics/btaa230
29. R. Mall, A. Elbasir, H. Almeer, Z. Islam, P. R. Kolatkar, S. Chawla, et al., A modeling framework for embedding-based predictions for compound–viral protein activity, *Bioinformatics*, **37** (2021), 2544–2555. https://doi.org/10.1093/bioinformatics/btab130
30. C. C. Wang, C. D. Han, Q. Zhao, X. Chen, Circular RNAs and complex diseases: From experimental results to computational models, *Briefings Bioinf.*, **22** (2021), bbab286. https://doi.org/10.1093/bib/bbab286
31. F. Y. Sun, J. Q. Sun, Q, Zhao, A deep learning method for predicting metabolite–disease associations via graph neural network, *Briefings Bioinf.*, **23** (2022), bbac266. https://doi.org/10.1093/bib/bbac266
32. Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, *IEEE Trans. Pattern Anal. Mach. Intell.*, **35** (2013), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50
33. W. Liu, H. Lin, L. Huang, L. Peng, T. Tang, Q. Zhao, et al., Identification of miRNA-disease associations via deep forest ensemble learning based on autoencoder, *Briefings Bioinf.*, **23** (2022), bbac104. https://doi.org/10.1093/bib/bbac104
34. U. Michelucci, An introduction to autoencoders, preprint, arXiv:2201.03898. https://doi.org/10.48550/arXiv.2201.03898
35. A. Goyal, A. Lamb, Y. Zhang, S. Z. Zhang, A. Courville, Y. Bengio, Professor forcing: A new algorithm for training recurrent networks, in *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, **29** (2016), 1–9.
36. T. Q. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, (2016), 785–794. https://doi.org/10.1145/2939672.2939785

37. L. Breiman, Random forests, *Mach. Learn.*, **45** (2001), 5–32. https://doi.org/10.1023/A:1010933404324
38. H. Zhang, The optimality of naive Bayes, *Aa*, **1** (2004), 3.
39. C. Cortes, V. Vapnik, Support-vector networks, *Mach. Learn.*, **20** (1995), 273–297. https://doi.org/10.1007/BF00994018
40. J. Laaksonen, E. Oja, Classification with learning k-nearest neighbors, in *Proceedings of International Conference on Neural Networks (ICNN'96)*, **3** (1996), 1480–1483.
41. Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time series, *Handb. Brain Theory Neural Networks*, **3361** (1995), 1995.
42. M. Wang, F. Y. Li, H. Wu, Q. Z. Liu, S. Q. Li, PredPromoter-MF (2L): A novel approach of promoter prediction based on multi-source feature fusion and deep forest, *Interdiscip. Sci.*, **14** (2022), 1–15. https://doi.org/10.1007/s12539-022-00520-4

AIMS Press