Computational Biology and Bioinformatics
2020; 8(1): 15-19
http://www.sciencepublishinggroup.com/j/cbb
doi: 10.11648/j.cbb.20200801.13
ISSN: 2330-8265 (Print); ISSN: 2330-8281 (Online)

Discovering *Escherichia coli* K-12 Promoter Features Using Convolutional Neural Network

Mengmeng Zhang, Lu Wang, Ping Wan*

College of Life Sciences, Capital Normal University, Beijing, China

Email address:
wanping@cnu.edu.cn (Ping Wan)

*Corresponding author

To cite this article:
Mengmeng Zhang, Lu Wang, Ping Wan. Discovering *Escherichia coli* K-12 Promoter Features Using Convolutional Neural Network. *Computational Biology and Bioinformatics*. Vol. 8, No. 1, 2020, pp. 15-19. doi: 10.11648/j.cbb.20200801.13

Received: May 24, 2020; **Accepted**: June 8, 2020; **Published**: June 20, 2020

Abstract: The mechanism of prokaryotic gene expression remains incompletely understood. Promoters are regions in genome that locating upstream to genes and regulate of gene expressions. Despite more and more *E. coli* K-12 promoter sequences have been obtained experimentally, and some regions such as -10 region and -30 region have been described, the features in promoter sequences are far from explicitly characterized. Here, we address this challenge using an approach based on the deep convolutional neural network (CNN). We collected six classes of *E. coli* K-12 promoter sequences which are all annotated as with strong evidence and belong to only one promoter class in RegulonDB database. Then, we applied the CNN model to recognize the six classes of promoters. The CNN model achieved an accuracy of above 97% for all six classes of promoters. Next, we extracted the weight matrix of the last convolution layer in CNN with the Grad-Cam algorithm, and convert the weight matrix to an information content matrix. Finally, we visualized the information content matrix as promoter logos using the logomaker tool and discover the promoter features in six classes of promoters. Our approach could not only find the previous described promoter feature regions, but could also discover promoter features with better sensitivity and accuracy. We provide a novel computational approach to discover features in biological sequences.

Keywords: Convolution Neural Network (CNN), Promoter, Biological Sequence, Features

1. Introduction

Promoters are regions of DNA that locating upstream to genes and regulate of gene expressions [1, 2]. In bacteria, the promoter is recognized by RNA polymerase and an associatedσfactor. In *E.coli*, seven classes of σ promoters have been found: σ24, σ28, σ32, σ38, σ54, σ70 and σ19 [3, 4]. RegulonDB is a database of the regulatory network of gene expression in *E. coli* K-12. Currently, RegulonDB has collected about 8000 *E. coli* K-12 promoter sequences. Among them, about 1200 promoter sequences having strong evidence being annotated belong to one or more σ classes [5].

Convolutional neural network (CNN) is one of the most important model in deep learning [6, 7]. CNNs have become the gold standard for numerous image analysis tasks [8]. It surpasses many established algorithms, such as support vector machines or random forests [9-11]. CNN also demonstrates a better performance in recognition of *E. coli* K-12 promoters than that of PSSM (Position-Specific Scoring Matrix) method [12-15]. In previous study, we have demonstrated that CNN outperforms PSSM method in identification of different promoter classes. However, it was unclear why CNN performs better [16].

The weight matrix of the last convolution layer in CNN contains the features extracted from the input data, and the Grad-Cam algorithm has realized the visualization for the weight matrices of CNN intermediate convolution layers [17]. In this work, we first train a CNN model, then we used the Grad-Cam algorithm to obtain the weight matrix of the last convolution layer. Further, we converted the matrix to an information content matrix. Finally, we used the logomaker tool [18] to visualize the information content matrix as a promoter logo. Our method not only successively displays the well-known -10 and -30 regions shown by the Weblogo method [19], but also be more accurate than the Weblogo result. Moreover, our method is more sensitive in discovering the dominant positions and bases in the promoter sequence

other than -10 and -30 regions. These factors contribute the CNN a better performance than PSSM in discovering promoter features.

2. Data & Methods

2.1. Promoter Sequences

The *E. coli* K-12 promoter DNA sequences were derived from RegulonDB database (http://regulondb.ccg.unam.mx/menu/download/datasets/index.jsp). We collected six classes of promoter sequences: σ24 (66), σ28 (10), σ32 (51), σ38 (102), σ54 (19) and σ70 (766) (The number in the parenthesis indicates the number of sequences). In RegulonDB database, these promoter sequences are all annotated as with strong evidence and belong to only one promoter class. The length of each promoter sequences is 81nt, including 1nt transcription start site (position 0), 60nt upstream region and 20nt downstream region (Figure 1A). In this study, we chose the 60nt upstream region as the dataset for CNN input.

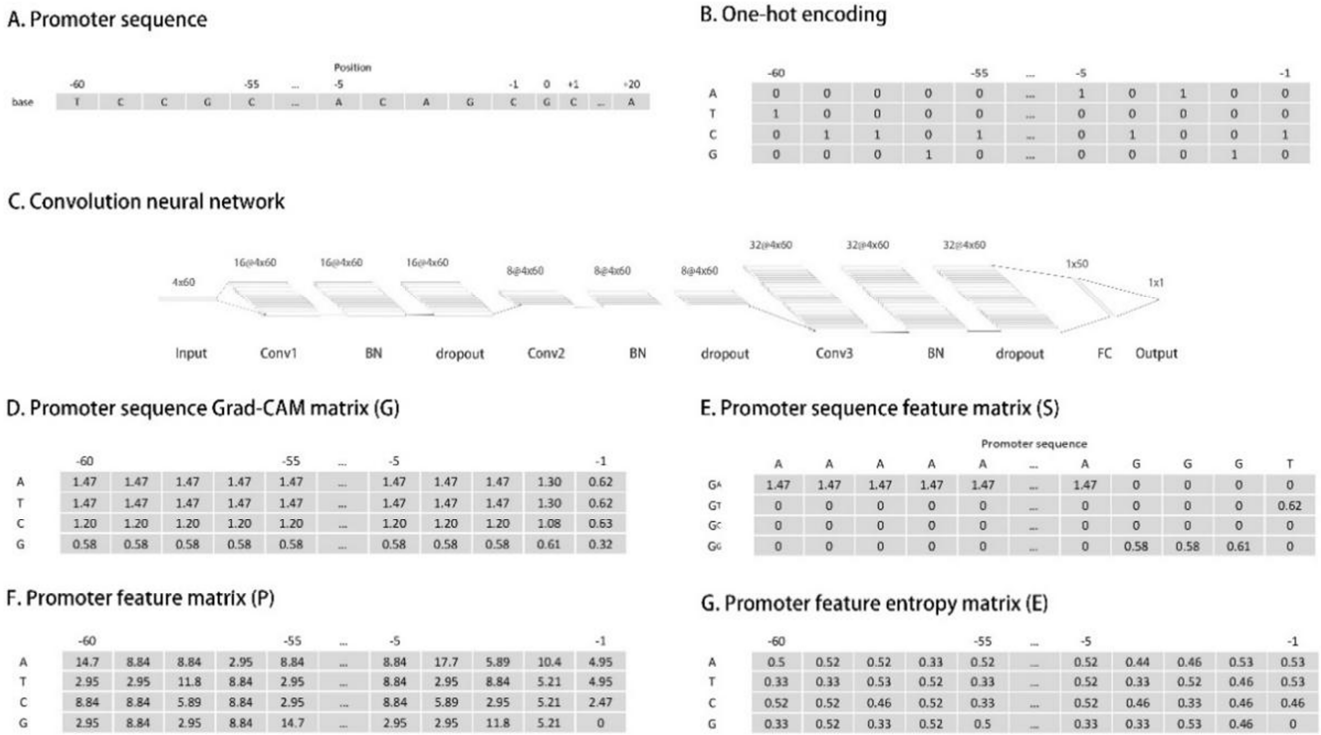

Figure 1. *Overview of the method.*

2.2. Convolution Neural Network

Since the CNN requires a two-dimension matrix as input, we first transformed the one-dimension promoter sequences into two-dimension matrices using one-hot encoding method. In detail, we encode each base into a four-digit list. In detail, A = [1,0,0,0], T = [0,1,0,0], C = [0,0,1,0], G = [0,0,0,1]. Then, the one-dimension promoter sequences are transformed into 4×60 two-dimension matrices (Figure 1B).

Next, we constructed a convolutional neural network. The CNN contains three convolution layers. Each convolution layer followed a batch normalization layer and a dropout layer to reduce the overfitting. We set the padding parameter "same" to keep the size of the last convolution layer matrix being the same as the input matrix. The last convolution layer is followed by one flatten layer and an output layer (Figure 1C).

We performed the 10-fold cross-validation to train the CNN model. We saved the model weights in an h5 file after each round of training and applied the weights from the last round as the starting weights for the next round of training. After several rounds of iterations the performance of the CNN was not improved any more. We used accuracy (Acc), specificity (Spec), sensitivity (Sen) and ROC curve to evaluate the performance of the CNN model.

Concretely:

Acc = (TP+TN) / (TP+TN+FP+FN)

Spec = TN / (FP+TN)

Sen = TP / (TP+FN)

Where: TP, TN, FP and FN are shorts for true positive, true negative, false positive, and false negative, respectively.

2.3. Obtaining the Last Convolution Layer Matrix (G) Using Grad-CAM Technique

The last convolution layer matrix contains the features extracted from the input matrix. For each promoter sequence, we used the Grad-CAM technique [17] to obtain its last convolution layer matrix (G) (Figure 1D).

2.4. Generating the Promoter Sequence Feature Matrix (S)

In matrix G, the row item represents four bases. While in a

particular promoter sequence, only one base occurs in one position, so we generated the promoter sequence feature matrix (S) for each promoter sequence from the matrix G (Figure 1E). In detail:

$$\mathbf{S}_{ij} = \begin{cases} \mathbf{G}_{ij} & if \quad base_{\mathbf{G}_{ij}} = base_{\mathbf{S}_j} \\ 0 & otherwise \end{cases} \tag{1}$$

2.5. Generating the Promoter Feature Matrix (P)

After we obtained the matrices **S** for each promoter sequences in six promoter classes, we created the promoter feature matrix P for each promoter class (Figure 1F). We calculated the P matrix as following:

$$\mathbf{P}_{ij} = \sum_{n=0}^{m} \mathbf{S}_{ij}^{n} \tag{2}$$

Where *m* is the number of sequences for a particular promoter class.

2.6. Generating the Promoter Feature Entropy Matrix (E)

Finally, we transform the promoter feature matrix P into the promoter feature entropy matrix E (Figure 1G) as following:

$$\mathbf{E}_{ij} = -\frac{\mathbf{P}_i^j}{\sum \mathbf{P}_i^j} \log_2 \frac{\mathbf{P}_i^j}{\sum \mathbf{P}_i^j} \tag{3}$$

Where P_i^j is the element of row *i* of *j* column in the matrix P.

2.7. Creating Promoter Logo

We use the logomaker [18] to visualize the promoter feature entropy matrix E and created promoter logos.

3. Result and Discussions

3.1. Identification of Promoters with CNN

First, we identified promoters with CNN. Table 1 shows that the accuracies of the CNN model for recognizing six promoter classes are all above 97%, while the AUCs (Area Under Curve) in ROC curves for six promoter classes are above 0.84 except sigma 38 (AUC=0.63) (Figure 2). The good performances of CNN in identifications of promoters are guarantees to discover features in promoters.

Table 1. *The performance of CNN in promoter identification.*

Promoter	Accuracy (%)	Sensitivity	Specificity	AUC
σ24	99.8	0.71	0.66	0.87
σ28	99.9	0.94	0.53	0.92
σ32	99.5	0.71	0.69	0.92
σ38	99.0	0.68	0.66	0.63
σ54	99.9	0.85	0.65	1.00
σ70	97.9	0.47	0.78	0.84

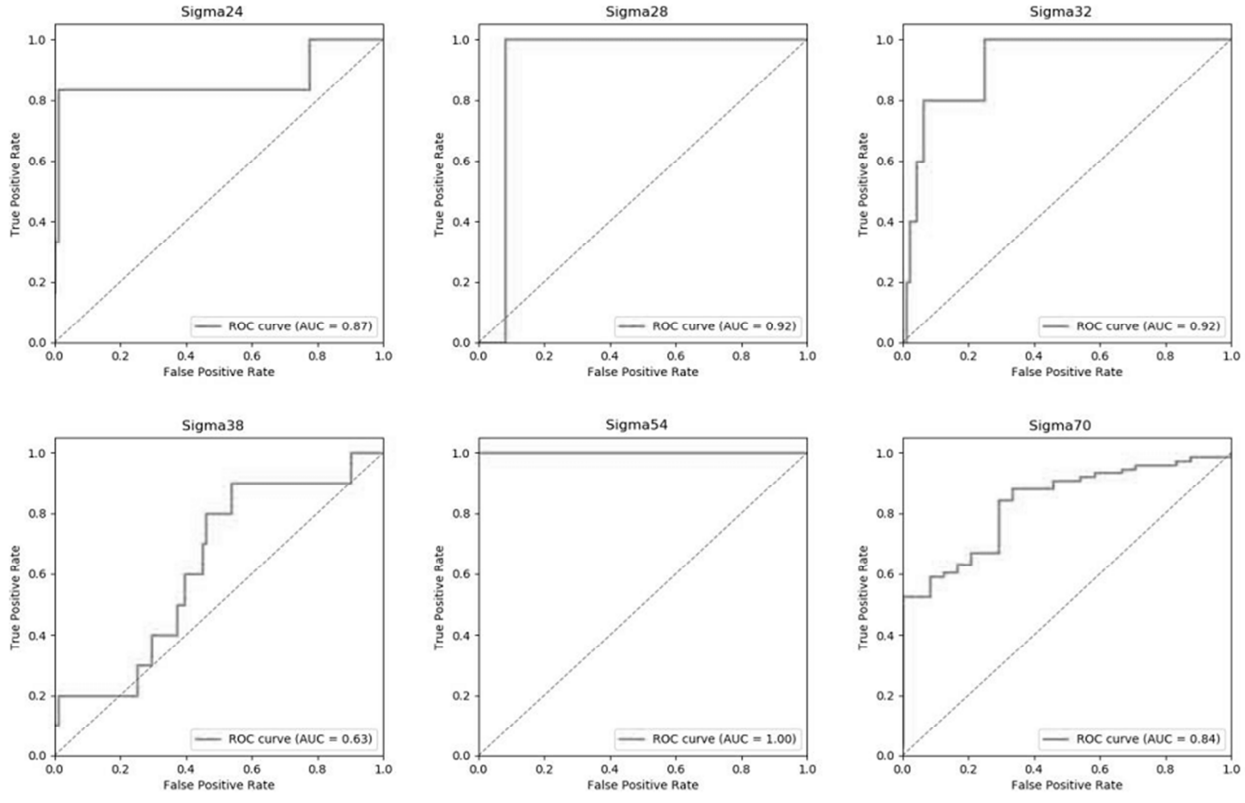

Figure 2. *ROC curves for promoter identifications using CNN.*

3.2. Promoter Features

We use the Grad-CAM technique [17] to extract the weight matrix (G) in the last convolution layer of CNN (Figure 1D), and then transform the matrix G into the promoter sequence feature matrix (S) (Figure 1E), the promoter feature matrix (P) (Figure 1F), and the promoter feature entropy matrix (E) (Figure 1G) in turn. The matrix E contains the promoter features in term of the information content. Finally, we visualize the matrix E using the logomaker tool [18]. Figure 3 shows the feature logos for six classes of promoters.

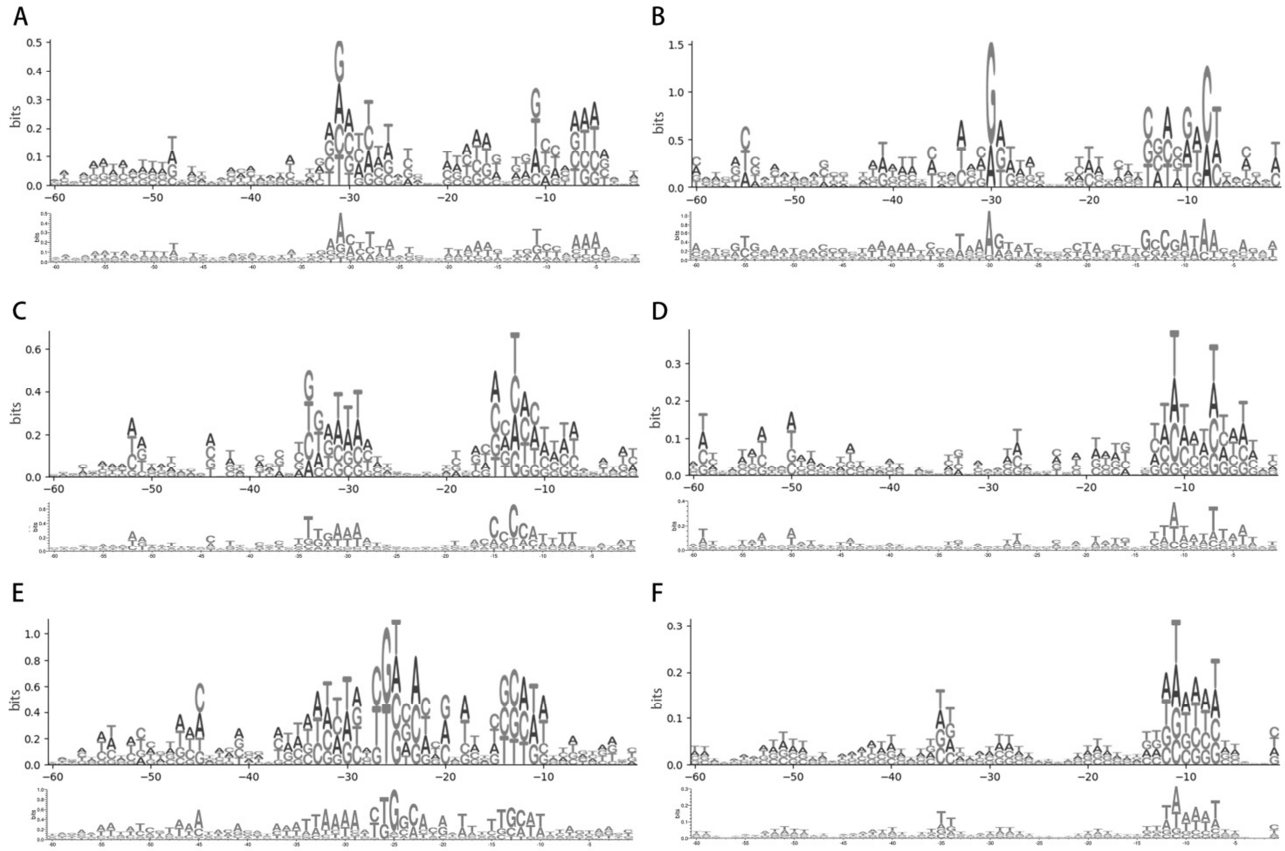

Figure 3. *Promoter logos. A. σ24, B. σ28, C. σ32, D. σ38, E. σ54, F. σ70. In each sub graph, the upper logo is created by the logomaker tool, the bottom one is created by the Weblogo tool.*

Figure 3 shows that our method could discover all feature regions successively found by Weblogo method (Figure 3). For example, the -10 region and the -30 region.

The Weblogo method is based on the probability of a base occurring at a position in the promoter sequence [19]. In detail, the Weblogo method finally generated a PSSM (Position-Specific Scoring Matrix) and visualized the PSSM. In previous study, we have demonstrated that CNN outperforms PSSM in promoter identification. An interesting question is why CNN performs better than PSSM? In this study, we found that CNN could discover the importance of each base at each position in the promoter sequence more precisely the PSSM. For example, in Figure 3B, at position -30, Weblogo shows that A is the dominant base, while our method shows that both A and G are important, and G is the dominant base. Moreover, our method is more sensitive than the PSSM method. For example, in Figure 3D, Weblogo shows faint signals outside the -10 region, but our method gives more signal details. The better sensitivity and accuracy contribute the CNN outperforming the PSSM method.

4. Conclusions

In this study, we demonstrated that deep convolutional neural network model performs better than the traditional bioinformatic algorithm in finding features in DNA sequences. The approch could also be applied in finding features in protein amino acid sequences.

Acknowledgements

This study was funded by the scientific research project of Beijing Municipal Commission of education, KM201610028010.

References

[1] He W, Jia C, Duan Y, *et al*. 70ProPred: a predictor for discovering sigma70 promoters based on combining multiple features. [J] BMC Systems Biology, 2018, 12 (4): 44.

[2] Barrios H, Valderrama B, Morett E. Compilation and analysis of sigma (54)-dependent promoter sequences. [J] Nucleic Acids Research, 1999, 27 (22): 4305-4313.

[3] Gruber TM, Gross CA. Multiple sigma subunits and the partitioning of bacterial transcription space. [J] Annual Review of Microbiology, 2003, 57: 441–66.

[4] Kang JG, Hahn MY, Ishihama A, Roe JH. Identification of sigma factors for growth phase-related promoter selectivity of RNA polymerases from Streptomyces coelicolor A3 (2). [J] Nucleic Acids Research, 1997, 25 (13): 2566-73.

[5] Santos-Zavaleta A, Salgado H, Gama-Castro S, *et al*. RegulonDB v 10.5: tackling challenges to unify classic and high throughput knowledge of gene regulation in *E. coli* K-12. [J] Nucleic acids research, 2019, 47: D212-D220.

[6] Lecun Y L, Bottou L, Bengio Y, *et al*. Gradient-based learning applied to document recognition. [J] Proceedings of the IEEE, 1998, 86 (11): 2278-2324.

[7] Lecun Y, Boser B, Denker J, *et al*. Backpropagation Applied to Handwritten Zip Code Recognition. [J] Neural Computation, 2014, 1 (4): 541-551.

[8] Alipanahi B, Delong A, Weirauch MT, *et al*. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. [J] Nature biotechnology, 2015, 33, 831.

[9] Zhou J, Troyanskaya OG. Predicting effects of noncoding variants with deep learning–based sequence model. [J] Nature methods, 2015, 12: 931.

[10] Kelley DR, Snoek J, Rinn JL. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. [J] Genome research, 2016, 26: 990-999.

[11] Eraslan G, Avsec Ž, Gagneur J, *et al*. Deep learning: new computational modelling techniques for genomics. [J] Nature Reviews Genetics, 2019, 20: 389-403.

[12] Gershenzon NI, Stormo GD, Ioshikhes IP. Computational technique for improvement of the position-weight matrices for the DNA/protein binding sites. [J] Nucleic Acids Research, 2005, 33 (7): 2290-301.

[13] Zhang L, Luo L. Splice site prediction with quadratic discriminant analysis using diversity measure. [J] Nucleic Acids Research, 2003, 31 (21): 6214-6220.

[14] Drioli S, Felluga F, Forzato C, *et al*. The recognition and prediction of σ 70, promoters in *Escherichia coli* K-12. [J] Journal of Theoretical Biology, 2006, 242 (1): 135.

[15] Gordon JJ, Towsey MW, Hogan JM, *et al*. Improved prediction of bacterial transcription start sites. [J] Bioinformatics, 2006, 22 (2): 142-148.

[16] Wang L, Wan P. Prediction of *Escherichia Coli* K-12 Promoters Using Convolutional Neural Network. [J] Computational Biology and Bioinformatics, 2018, 6: 2.

[17] Selvaraju RR, Cogswell M, Das A, *et al*. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. arXiv: 1610.02391, 2019, DOI: 10.1007/s11263-019-01228-7.

[18] Tareen A, Kinney JB. Logomaker: Beautiful sequence logos in python. [J] Bioinformatics, 2020, 36 (7): 2272–2274.

[19] Crooks GE, Hon G, Chandonia JM, *et al*. WebLogo: a sequence logo generator. [J] Genome research, 2004, 14: 1188-1190.