genes

MDPI

Article

PromoterLCNN: A Light CNN-Based Promoter Prediction and Classification Model

Daryl Hernández [1,†], Nicolás Jara [1,*,†], Mauricio Araya [1], Roberto E. Durán [2] and Carlos Buil-Aranda [3]

[1] Department of Electronics Engineering, Universidad Técnica Federico Santa María, Valparaiso 2390123, Chile; daryl.hernandez@usm.cl (D.H.); mauricio.araya@usm.cl (M.A.)
[2] Laboratorio de Microbiología Molecular y Biotecnología Ambiental, Departamento de Química & Centro de Biotecnología Daniel Alkalay Lowitt, Universidad Técnica Federico Santa María, Valparaiso 2390123, Chile; ro.duran.vargas@gmail.com
[3] Department of Informatics Engineering, Universidad Técnica Federico Santa María, Valparaiso 2390123, Chile; cbuil@inf.utfsm.cl
[*] Correspondence: nicolas.jara@usm.cl
[†] These authors contributed equally to this work.

Abstract: Promoter identification is a fundamental step in understanding bacterial gene regulation mechanisms. However, accurate and fast classification of bacterial promoters continues to be challenging. New methods based on deep convolutional networks have been applied to identify and classify bacterial promoters recognized by sigma (σ) factors and RNA polymerase subunits which increase affinity to specific DNA sequences to modulate transcription and respond to nutritional or environmental changes. This work presents a new multiclass promoter prediction model by using convolutional neural networks (CNNs), denoted as PromoterLCNN, which classifies *Escherichia coli* promoters into subclasses σ^{70}, σ^{24}, σ^{32}, σ^{38}, σ^{28}, and σ^{54}. We present a light, fast, and simple two-stage multiclass CNN architecture for promoter identification and classification. Training and testing were performed on a benchmark dataset, part of RegulonDB. Comparative performance of PromoterLCNN against other CNN-based classifiers using four parameters (Acc, Sn, Sp, MCC) resulted in similar or better performance than those that commonly use cascade architecture, reducing time by approximately 30–90% for training, prediction, and hyperparameter optimization without compromising classification quality.

Keywords: bacterial promoters; convolutional neural networks; bioinformatics; deep learning; PromoterLCNN

check for updates

Citation: Hernández, D.; Jara, N.; Araya, M.; Durán, R.E.; Buil-Aranda, C. PromoterLCNN: A Light CNN-Based Promoter Prediction and Classification Model. *Genes* **2022**, *13*, 1126. https://doi.org/10.3390/genes13071126

Academic Editor: Piero Fariselli

Received: 23 May 2022
Accepted: 20 June 2022
Published: 23 June 2022

1. Introduction

Bacterial promoters are DNA sequences positioned upstream of the transcription start site (TSS), crucial for recognition by the RNA polymerase (RNAP) [1]. Promoters initiate transcription due to their affinity to the RNAP, and therefore are essential for maintaining cellular homeostasis. Promoter affinity is highly determined by two conserved hexamers in positions -10 and -35 upstream of the TSS [1,2]. The promoter efficiency is also modulated by a spacer region around 17 ± 3 bp length between the hexamers and the adjacent nucleotide sequences (UP element, extended -10 element; (Figure 1A) with lower conservancy in comparison to the -10 and -35 promoter elements [3,4].

The bacterial RNAP consists of six subunits—the core RNAP composed of five subunits ($\alpha_2\beta\beta'\omega$), and the sigma ($\sigma$) subunit or σ factor (Figure 1B). The σ factor reversibly binds to the core RNAP modulating the DNA-binding characteristics of the enzyme, increasing the affinity for promoters [1]. The most abundant sigma factor is σ^{70} (σ^A), encoded by *rpoD* and responsible for almost all gene expression. Different σ factors have been identified, each one competing for the core RNAP and initiating the transcription of different genes associated to a specific nutritional status or environmental condition [1,5]. Alternative sigma factors targeting different promoters have been characterized in *E. coli*. These include the σ^H (σ^{32}) encoded by *rpoH* involved in heat shock stress response, the σ^S (σ^{38}) associated to stationary phase regulation, σ^E (σ^{24}) related to extracytoplasmatic functions, σ^N (σ^{54}) relevant to nitrogen metabolism, and σ^F (σ^{28}) associated to flagellar synthesis and chemotaxis [5]. Each σ factor recognizes and facilitates the binding of RNAP to different promoters with distinct consensus sequences, which hampers the in silico identification of promoter sequences in bacterial genomes [6].

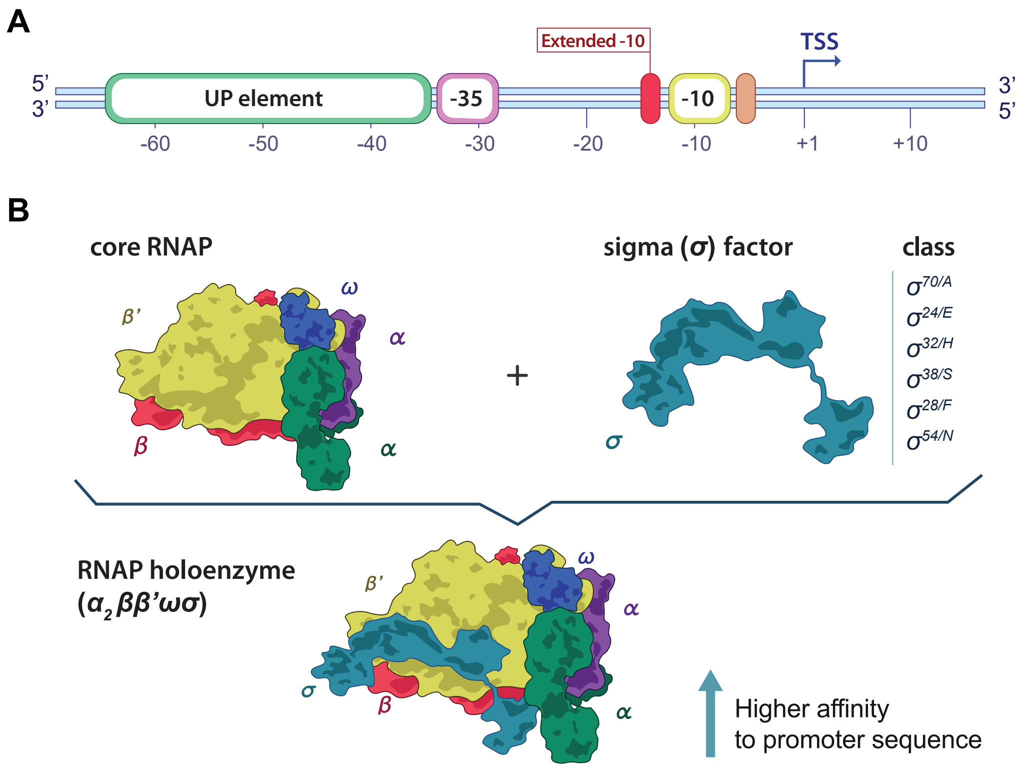


Figure 1. Graphic. representation of the bacterial DNA promoter and the RNA polymerase (RNAP) holoenzyme. (**A**) Key elements of the bacterial DNA promoter are crucial for RNAP affinity and binding. The coloured rounded rectangles represent key elements of the bacterial promoter. These include the UP element, the −35 and −10 hexamers, the extended −10 element, and the discriminator sequence. The transcriptional start site (TSS) is represented at the +1 position. (**B**) The RNAP holoenzyme components and subunits. The two components of the functional RNAP complex are illustrated in the upper part: the core RNAP enzyme, composed of the α, β, and ω subunits; and the sigma (σ) factor, which improves the RNAP–DNA affinity for transcription. At the bottom, the binding of the σ factor to the core enzyme completes the RNAP holoenzyme ($\alpha_2\beta\beta'\omega\sigma$).

Accurate and fast prediction of promoter sites associated with a σ factor remains a troublesome issue in genomics and molecular biology, despite being highly relevant for gene expression patterns, genetic regulatory networks, and synthetic biology studies, wherein the synthesis of new nucleotide sequences can incorporate undesired promoter sequences [6–8]. Traditional methods carried out by low-scale methods (e.g., DNA footprinting, primer extension, electrophoretic mobility shift assay) are slow and time-consuming. Meanwhile, even the use of high-throughput technologies (e.g., RNA-seq, systematic genomic evolution of ligands by exponential enrichment) [9–11] still does not compare to the explosive amount of genomic data generated during the last decade [12,13]. In silico prediction based on bioinformatic tools has also been explored based on sequence information, first as position weight-matrices [14–16], and later by using machine-learning (ML) techniques. The latter stand out as they do not require manual assembly of the characteristics or patterns to be detected, are free-alignment methods that do not require comparison with known sequences or databases, and work on raw information [6,7,17–22]. This feature is particularly relevant because one of the biggest problems identifying promoters is precisely their high variability. Then, depending on the overall context, a sequence may or may not be part of a promoter, as the so-called *Pribnow box* conformed by TATAAT (the -10 promoter element) is typically associated with promoters, but their sole identification does not define them.

To date, there are still few proposals based on neural networks for promoter sequence classification capable of making predictions with specificity and sensitivity values more significant than 80% and even 90%. These techniques generally use data from promoters recognized by σ^{70} factor [7,20]. In bacteria, they have been tested on the model strains (*E. coli* str. K-12 substr. MG1655 and *Bacillus subtilis* subsp. *subtilis* str. 168) by using already characterized promoter sequences as positive samples and delivering random sequences of coding sequences (CDS) as the negative sample. These methods generally take the problem as a classification problem, classifying the sequence as either a promoter σ^{70} or a non-promoter σ^{70} [6,7,20].

In recent years, various strategies have been explored to address the problem described, emphasizing those based on machine-learning techniques. Notably, the following references are based on convolutional neural networks (CNN). Reference [20] proposed CNNProm, consisting of a CNN of a one-dimensional (1D) convolution layer followed by a max-pooling and a fully connected ReLU, with sigmoid-triggered output. Its dataset contains bacterial, human, mouse, and plant sequences, each of 81 nucleotides for bacteria and 251 for the rest. These were coded according to one-hot encoding: $A \rightarrow (1,0,0,0,0)$, $T \rightarrow (0,1,0,0,0)$, $G \rightarrow (0,0,1,0)$, $C \rightarrow (0,0,0,1)$. For negative examples, random coding sequences were selected. Qian et al., proposed improvements to CNNProm [19]. They used support vector machines to highlight the importance of element sequences of eukaryotic promoters (9 elements included), compressing the non-element sequences of the promoter. The promoter sequences also used one-hot encoding. Subsequently, [7]. proposed DeePromoter, adding a long short-term memory (LSTM) to the architecture. They also stop using coding sequences as negative examples, generating them from the positive ones and replacing random parts, increasing the method robustness against false positives. Another novelty is incorporating dropout layers to increase robustness and prevent overfitting. pcPromoter-CNN [22] presents a convolutional neural network model for promoter prediction and classification of sigma sub-classes following a cascading architecture, performing a binary classification. First, classifying promoters and non-promoters, then for the promoters, it checks if it belongs to σ^{70}; if not, it continues with σ^{24} and so on. Finally, Ref. [18] developed IPromoter-BnCNN, capable of classifying promoters into five sigma categories by using a series of cascading binary classifiers as well as pcPromoters-CNN. In IPromoter-BnCNN, each binary classifier is a CNN of 4 parallel branches.

This study proposes a light two-stage promoter prediction and classification model by using multiclass CNN. We denote this model as PromoterLCNN. The first stage was designed to distinguish between promoters and non-promoters, and the second stage performs the sigma classification by using a multiclass classification model. As stated by previous works in the literature such as [22–24], we use Chou et al.'s 2011 [25] five-step rules for a clear presentation and validation of the model. We applied the model to *E. coli* by using RegulonDB v9.3 and 10.7 benchmark databases [26,27], for training and independent testing, respectively. We validate the results by using the K-fold cross-validation technique, examining four performance evaluation metrics to compare them with the best-performing methods found in the literature.

This paper is structured as follows. Section 2 presents our prediction and classification model. Section 3 presents the numerical results and a further discussion. Lastly, Section 4 states several conclusions and final remarks.

2. Inputs and Methods

In this section, we present the PromoterLCNN model, following the five-steps rule suggested by [25]. Figure 2 presents an overview of our PromoterLCNN model, and each of the five steps is described in detail as follows.

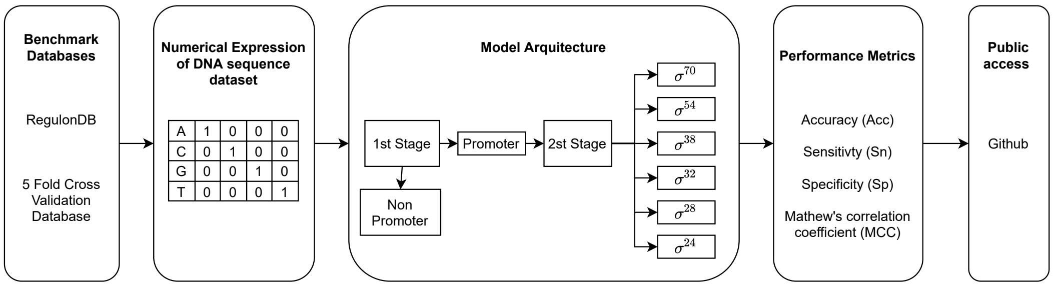


Figure 2. Overview of the PromoterLCNN model.

2.1. Benchmark and Test Datasets

Following the path set by previous works [7,18,22], the first essential action for generating a valid statistical promoter predictor is selecting a suitable benchmark dataset for training the model and an independent dataset for testing. We use the same disjointed datasets as in the literature [22–24], the benchmark based on experimentally verified promoter sequences and non-promoter ones extracted from coding zones. They are part of RegulonDB version 9.3, in which each entry has 81 bp [26].

The independent test dataset comprises promoter samples only, extracted from RegulonDB version 10.7, which are also experimentally verified [27].

The Training and Test Dataset sample size can be found in Table 1. The entire dataset $\mathcal{D}$ is composed of two subsets—the $\mathcal{P}$ promoters and the $\overline{P}$ Non-promoter—as stated in Equation (1):

$$\mathcal{D} = \mathcal{P} \cup \overline{P}. \tag{1}$$

Similarly, the P promoter subset contains all sigma sub-classes. Promoter sequences classified as σ^{24}, σ^{28}, σ^{32}, σ^{38}, σ^{54}, and σ^{70} according to the affinity to each σ factor is.

Accordingly, the P promoter subset can be defined as follow in Equation (2):

$$\mathcal{P} = \sigma^{24} \cup \sigma^{28} \cup \sigma^{32} \cup \sigma^{38} \cup \sigma^{54} \cup \sigma^{70}. \tag{2}$$

Table 1. Training and test dataset sample sizes.

Classes	Training Dataset	Test Dataset
Promoter ($\mathcal{P}$)	2860	256
Non-Promoter ($\overline{P}$)	2860	0
σ^{70}-Promoter	1694	199
σ^{24}-Promoter	484	30
σ^{32}-Promoter	291	13
σ^{38}-Promoter	163	10
σ^{28}-Promoter	134	4
σ^{54}-Promoter	94	0

2.2. Mathematical Formulation of DNA Sequence

The DNA sequence comprises a large string mixing four nucleotides denoted as A, C, G, and T. Similar to previous works, we use one-hot encoding to transform the DNA sequences to a binary form [20,22]. Each nucleotide is converted to a four-element vector with a single element value equal to one, and all other element values are 0. All numerical representation of nucleotides are as follows:

$$A \rightarrow (1,0,0,0)$$
$$C \rightarrow (0,1,0,0)$$
$$G \rightarrow (0,0,1,0)$$
$$T \rightarrow (0,0,0,1)$$

2.3. Model Architecture

The prediction and classification model denoted as PromoterLCNN is composed of two stages. The first stage recognises the DNA sequence as a promoter or not a promoter. Next, the second stage serves as a multiclass classification layer for the acknowledged promoters of the previous stage, identifying the sigma subclass associated with the DNA sequence. Both stages are CNN architectures. We trained the first stage of the model by using the promoter and non-promoter benchmark database ($\mathcal{D}$), and for the second stage, we trained the multiclass model by using all sigma promoters benchmark database ($\mathcal{P}$) displayed in Table 1.

Next, further details of each stage are explained.

2.3.1. First Stage

As previously mentioned, the first stage focuses on the binary recognition of promoters and non-promoters. To face this endeavour, we design an architecture based on convolutional neural networks, illustrated in Figure 3.

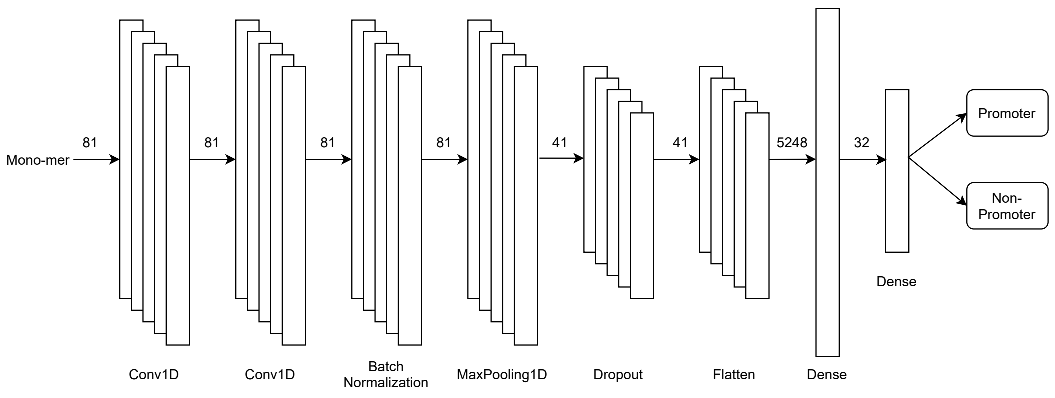


Figure 3. First-stage multiclass CNN architecture for promoters recognition in PromoterLCNN.

This stage of the architecture consists of 2 single-dimensional convolution layers in tandem, followed by batch normalisation, max-pooling, and dropout, preventing overfitting. Later, the features obtained by the model are flattened on a flatten layer. Finally, the set passes through two fully connected dense layers for the binary classification.

2.3.2. Second Stage

Thereupon, the second stage performs a multiclass classification of promoter subclasses. This stage of the architecture uses the same first-stage convolutional neural network architecture with the sole difference that the possible classification exits are six instead of two. This stage of the architecture is displayed in Figure 4.

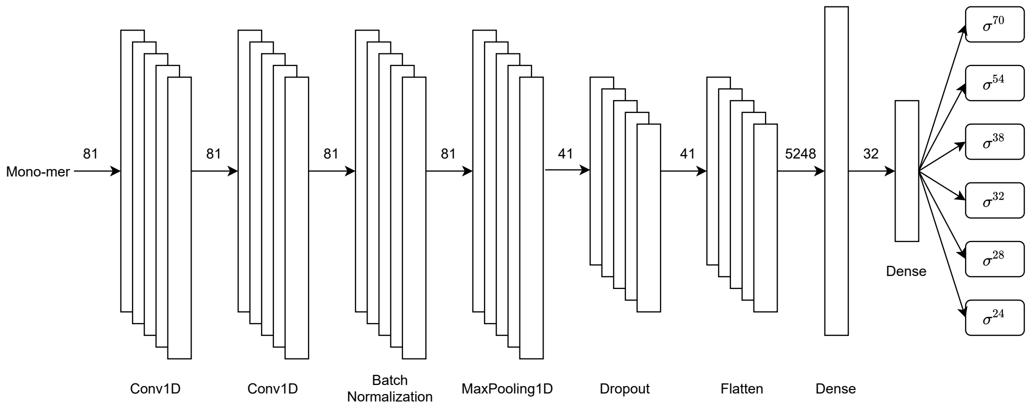


Figure 4. Second-stage multiclass CNN architecture for sigma classification in PromoterLCNN.

Therefore, this stage consists of two single-dimensional convolution layers. Next, batch normalisation, max-pooling, dropout, flatten, and finally, two fully connected dense layers are used for the multiclass classification of sigma sub-classes.

A hyperparameter tuning is performed to select the most acceptable parameters on all the layers in both stages' architectures. Next, we explain this tuning process.

2.3.3. Hyperparameter Tuning

For choosing the finest parameters for the two convolutions, pooling, dropout, and the last two dense layers, hyperparameter tuning is performed. Table 2 presents the candidate values of hyperparameters used for the hyperparameter optimization process.

Table 2. Hyperparameter tuning parameters and their candidate values.

Hyper Parameter	Candidate Values
Number of Convolution Filters	16, 32, 64, 128
Convolution Kernel Size	3, 5, 7, 9
Kernel Regularizers (L2)	$1\times 10^{-5}, 1 \times 10^{-4}, 1 \times 10^{-3}, 1 \times 10^{-2}$
Dropout Rate	$0.15, 0.20, \ldots, 0.50$
Dense Layer Neurons	$8, 16, \ldots, 64$

The hyperparameter tuning process was performed by Keras Tuner by using the Hyperband method. This process is time-consuming; thus, some selected hyperparameters were considered, as shown in Table 2.

In both stages, the first convolution layer uses 128 filters of kernel size 5 with 10^{-3} L2 kernel regularizer and 10^{-2} bias regularizer. The second convolutional layer uses 128 filters of kernel size 9, with 10^{-3} and 10^{-5} for the L2 kernel regularizer and bias regularizer, respectively. Both convolution layers use a ReLU activation function. The max-pooling layer uses a pool size of 2 and 2 strides. Then the dropout layer drops 45% of the features previously obtained. Finally, the two fully connected dense layers have 32 neurons using a ReLU activation function, with L2 kernel regularizer and bias regularizer values of 10^{-3}.

2.4. Performance Metrics

We use a straightforward strategy for measuring our model performance. We use the same strategy used by several works in the literature [18,22]. The quality of our prediction and classification model is defined by four metrics, *Acc*, *Sn*, *Sp*, and *MCC*, corresponding to the accuracy, sensitivity, specificity, and Matthews correlation coefficient, respectively. Their definitions are as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}, \quad (3)$$

$$Sn = \frac{TP}{TP + FN}, \tag{4}$$

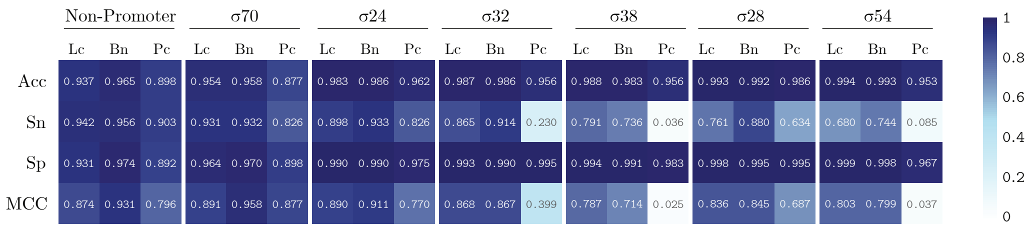
$$Sp = \frac{TN}{TN + FP}, \tag{5}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{6}$$

in which TP, TN, FP, and FN stand for the true positives, true negatives, false positives, and false negatives, respectively.

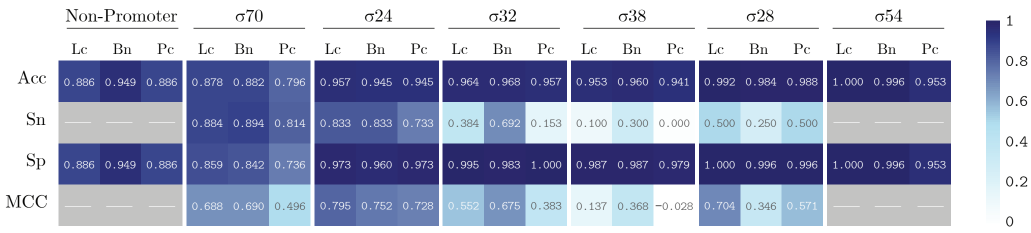
3. Results and Discussion

We train our model by using k-fold validation with a k value equal to 5. The performance results are summarized in Figures 5 and 6, illustrating the results for our PromoterLCNN (Lc), and obtaining performances better than or similar to pcPromoter-CNN (Pc) and iPromoter-BcNN (Bc) methods, measuring their accuracy (Acc), sensitivity (Sn), specificity (Sp) and Matthews correlation coefficient (MCC), for the training dataset (Figure 5) and the independent test dataset (Figure 6) mentioned in Section 2.1. We remark that these figures are also presented as a heatmap, in which the colour intensity indicates the quality of each value displayed here.



	Non-Promoter			σ70			σ24			σ32			σ38			σ28			σ54		
	Lc	Bn	Pc	Lc	Bn	Pc	Lc	Bn	Pc	Lc	Bn	Pc	Lc	Bn	Pc	Lc	Bn	Pc	Lc	Bn	Pc
Acc	0.937	0.965	0.898	0.954	0.958	0.877	0.983	0.986	0.962	0.987	0.986	0.956	0.988	0.983	0.956	0.993	0.992	0.986	0.994	0.993	0.953
Sn	0.942	0.956	0.903	0.931	0.932	0.826	0.898	0.933	0.826	0.865	0.914	0.230	0.791	0.736	0.036	0.761	0.880	0.634	0.680	0.744	0.085
Sp	0.931	0.974	0.892	0.964	0.970	0.898	0.990	0.990	0.975	0.993	0.990	0.995	0.994	0.991	0.983	0.998	0.995	0.995	0.999	0.998	0.967
MCC	0.874	0.931	0.796	0.891	0.958	0.877	0.890	0.911	0.770	0.868	0.867	0.399	0.787	0.714	0.025	0.836	0.845	0.687	0.803	0.799	0.037

Figure 5. Performance over the training dataset for PromoterLCNN (Lc), pcPromoter-CNN (Pc) and iPromoter-BcNN (Bc), measuring accuracy (Acc), sensitivity (Sn), specificity (Sp) and Matthews correlation coefficient (MCC).



	Non-Promoter			σ70			σ24			σ32			σ38			σ28			σ54		
	Lc	Bn	Pc	Lc	Bn	Pc	Lc	Bn	Pc	Lc	Bn	Pc	Lc	Bn	Pc	Lc	Bn	Pc	Lc	Bn	Pc
Acc	0.886	0.949	0.886	0.878	0.882	0.796	0.957	0.945	0.945	0.964	0.968	0.957	0.953	0.960	0.941	0.992	0.984	0.988	1.000	0.996	0.953
Sn	—	—	—	0.884	0.894	0.814	0.833	0.833	0.733	0.384	0.692	0.153	0.100	0.300	0.000	0.500	0.250	0.500	—	—	—
Sp	0.886	0.949	0.886	0.859	0.842	0.736	0.973	0.960	0.973	0.995	0.983	1.000	0.987	0.987	0.979	1.000	0.996	0.996	1.000	0.996	0.953
MCC	—	—	—	0.688	0.690	0.496	0.795	0.752	0.728	0.552	0.675	0.383	0.137	0.368	−0.028	0.704	0.346	0.571	—	—	—

Figure 6. Performance on the test dataset for PromoterLCNN (Lc), pcPromoter-CNN (Pc) and iPromoter-BcNN (Bc), measuring accuracy (Acc), sensitivity (Sn), specificity (Sp) and Matthews correlation coefficient (MCC).

As displayed in Figure 5, the results using the training database show that our approach (Lc) performs better than the pcPromoter-CNN (Pc) method for every metric and every promoter. In fact, a weighted average of accuracy with respect to the number of elements in each class gives 96.7% for Lc and 91.2% for Pc. From the sensitivity results, our method tends to produce a few more misclassifications than iPromoter-BcNN for this dataset, and the specificity row suggests that these corresponded to false-negative non-promoters as the dataset is balanced. In other words, the approach is very efficient detecting and classifying promoters, with minor contamination of non-promoters. In any case, the sensitivity values of the PromoterLCNN drop or improve a few points concerning iPromoter-BcNN, but it does not significantly fail like the pcPromoter-CNN does for σ^{38} and σ^{32}. The MCC shows that the confusion matrix quality for most classes is very similar for the two leading methods. A few misclassifications on the training data might indicate that the network is generalizing correctly, as opposed to overfitting problems found in shallow or classical learning methods. To verify this, we must achieve similar performance in the testing dataset.

Regarding the results over the test dataset presented in Figure 6, our approach outperforms the pcPromoter-CNN, as the weighted averages are 89.6% for Lc and 83.0% for Pc, and achieves comparable (and even slightly better) results than iPromoter-BcNN in terms of accuracy. PromoterLCNN produces similar or even fewer false negatives, as illustrated by the specificity values of promoters (Sp). However, it is important to recall Table 1 for a tempered analysis: each promoter class has only a few examples, and there are no non-promoters. Therefore, the most relevant statistics here are those obtained for σ^{70} and σ^{24} (i.e., $n \geq 30$). These statistics show that our approach is better at classifying promoters, at the cost of non-detecting a few of them (false negatives) compared to iPromoter-BcNN.

Establishing the competitive performance of our approach is vital to notice the parsimony in our proposal. We present an architecture appreciably lighter than pcPromoter-CNN and iPromoter-BcNN. The first one uses a cascading architecture for classifying first promoters and non-promoters, and later sigma sub-classes sequentially, with nine different layers each [22]. The second one uses the same cascading architecture, with four parallel layers (4 layers each branch), converging into 4 layers at the end [18]. On the contrary, our method has only two stages with eight different layers each. This feature significantly diminishes the computing time for training, prediction, and hyperparameter optimization processes, taking a tenth of the time compared to iPromoter-BcNN, and 30% less than pcPromoter-CNN.

4. Conclusions

This work presents a two-stage promoter prediction and classification model by using a multiclass convolution neural network called PromoterLCNN. The first stage of the architecture attempts to recognize between promoters and non-promoters, and the second stage engages the sigma classification by using a straightforward multiclass classification model, in contrast to standard approaches found in the literature. We use Chou et al.'s five-step rules for our model presentation and validation process by using *E. coli* databases found in RegulonDB v9.3 and v10.7 benchmark databases for training and independent testing, respectively.

We used a K-fold cross-validation training and assessed the results with an independent test dataset. We found out that our method outperforms one of the most competitive methods in the literature. For the more recent state-of-the-art approaches, our proposal has competitive results in accuracy, with better promoter-type classification. Remarkably, PromoterLCNN has a lighter architecture than other models, leading to a shorter time for the hyperparameter tuning, training, and prediction processes without compromising classification quality, an attractive quality for molecular or synthetic biologists working with nucleotide sequences on a daily basis. By using part of a genome or a newly synthetized sequence as input data, PromoterLCNN might help researchers and users working in the field of bacterial genomics, molecular biology, and bioinformatics to identify bacterial promoters and classify them into each of the σ subclasses validated in this study.

Author Contributions: Conceptualization, N.J., M.A., R.E.D. and C.B.-A.; methodology, N.J. and M.A.; software, D.H. and N.J.; validation, M.A., R.E.D. and C.B.-A.; formal analysis, D.H., N.J. and M.A.; investigation, D.H., N.J., M.A., R.E.D. and C.B.-A.; resources, D.H., R.E.D. and C.B.-A.; data curation, D.H.; writing—original draft preparation, D.H. and N.J.; writing—review and editing, N.J., M.A., R.E.D. and C.B.-A.; visualization, N.J.; supervision, N.J.; project administration, C.B.-A. and R.E.D.; funding acquisition, C.B.-A. and R.E.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by Project USM PI_M_2020_43 (D.H., N.J, M.A., R.E.D., C.B.-A.), Millennium Institute for Foundational Research on Data (IMFD) (C.B.-A.), ANID-Basal Project FB0008 (AC3E; M.A.) and ANID PIA/APOYO AFB180002 (CCTVal; M.A.).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: PromoterLCNN is available in the following github repository: https://github.com/occasumlux/Promoters (accessed on 1 June 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Helmann, J.D. Where to begin? Sigma factors and the selectivity of transcription initiation in bacteria. *Mol. Microbiol.* **2019**, *112*, 335–347. [CrossRef]
2. Bervoets, I.; Van Brempt, M.; Van Nerom, K.; Van Hove, B.; Maertens, J.; De Mey, M.; Charlier, D. A sigma factor toolbox for orthogonal gene expression in *Escherichia coli*. *Nucleic Acids Res.* **2018**, *46*, 2133–2144. [CrossRef]
3. Mitchell, J.E.; Zheng, D.; Busby, S.J.W.; Minchin, S.D. Identification and analysis of 'extended -10' promoters in *Escherichia coli*. *Nucleic Acids Res.* **2003**, *31*, 4689–4695. [CrossRef]
4. Typas, A.; Hengge, R. Differential ability of sigma(s) and sigma70 of *Escherichia coli* to utilize promoters containing half or full UP-element sites. *Mol. Microbiol.* **2005**, *55*, 250–260. [CrossRef]
5. Abril, A.G.; Rama, J.L.R.; Sánchez-Pérez, A.; Villa, T.G. Prokaryotic sigma factors and their transcriptional counterparts in Archaea and Eukarya. *Appl. Microbiol. Biotechnol.* **2020**, *104*, 4289–4302. [CrossRef]
6. Cassiano, M.H.A.; Silva-Rocha, R. Benchmarking bacterial promoter prediction tools: Potentialities and limitations. *mSystems* **2020**, *5*, 439. [CrossRef]
7. Oubounyt, M.; Louadi, Z.; Tayara, H.; Chong, K.T. DeePromoter: Robust Promoter Predictor Using Deep Learning. *Front. Genet.* **2019**, *10*, 286. [CrossRef]
8. Di Salvo, M.; Pinatel, E.; Talà, A.; Fondi, M.; Peano, C.; Alifano, P. G4PromFinder: An algorithm for predicting transcription promoters in GC-rich bacterial genomes based on AT-rich elements and G-quadruplex motifs. *BMC Bioinform.* **2018**, *19*, 36. [CrossRef]
9. Ishihama, A.; Shimada, T.; Yamazaki, Y. Transcription profile of *Escherichia coli*: Genomic SELEX search for regulatory targets of transcription factors. *Nucleic Acids Res.* **2016**, *44*, 2058–2074. [CrossRef]
10. Dahl, J.; Collas, P. A rapid micro chromatin immunoprecipitation assay (ChIP). *Nat. Protoc.* **2008**, *3*, 1032–1045. [CrossRef]
11. Matsumine, H.; Yamamura, Y.; Hattori, N.; Kobayashi, T.; Kitada, T.; Yoritaka, A.; Mizuno, Y. A Microdeletion of D6S305 in a Family of Autosomal Recessive Juvenile Parkinsonism (PARK2). *Genomics* **1998**, *49*, 143–146. [CrossRef]
12. Behjati, S.; Tarpey, P.S. What is next generation sequencing? *Arch. Dis. Child. Educ. Pract.* **2013**, *98*, 236–238. doi: 10.1136/archdischild-2013-304340. [CrossRef]
13. Zhang, J.; Chiodini, R.; Badr, A.; Zhang, G. The impact of next-generation sequencing on genomics. *J. Genet. Genom.* **2011**, *38*, 95–109. [CrossRef]
14. Solovyev, V.; Salamov, A. Automatic annotation of microbial genomes and metagenomic sequences. In *Metagenomics and Its Applications in Agriculture, Biomedicine and Environmental Studies*; Li, R.W., Ed.; Nova Science Publishers, Inc.: Hauppauge, NY, USA, 2011; pp. 61–78.
15. Wang, J.; Hannenhalli, S. A mammalian promoter model links cis elements to genetic networks. *Biochem. Biophys. Res. Commun.* **2006**, *347*, 166–177. [CrossRef]
16. Münch, R.; Hiller, K.; Grote, A.; Scheer, M.; Klein, J.; Schobert, M.; Jahn, D. Virtual Footprint and PRODORIC: An integrative framework for regulon prediction in prokaryotes. *Bioinformatics* **2005**, *21*, 4187–4189. [CrossRef]
17. He, W.; Jia, C.; Duan, Y.; Zou, Q. 70ProPred: A predictor for discovering sigma70 promoters based on combining multiple features. *BMC Syst. Biol.* **2018**, *12*, 44. [CrossRef]
18. Amin, R.; Rahman, C.R.; Ahmed, S.; Sifat, M.H.R.; Liton, M.N.K.; Rahman, M.M.; Khan, M.Z.H.; Shatabda, S. iPromoter-BnCNN: A novel branched CNN-based predictor for identifying and classifying sigma promoters. *Bioinformatics* **2020**, *36*, 4869–4875. [CrossRef]
19. Qian, Y.; Zhang, Y.; Guo, B.; Ye, S.; Wu, Y.; Zhang, J. An Improved Promoter Recognition Model Using Convolutional Neural Network. In Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 23–27 July 2018; Volume 1, pp. 471–476. [CrossRef]
20. Umarov, R.K.; Solovyev, V.V. Recognition of prokaryotic and eukaryotic promoters using convolutional deep learning neural networks. *PLoS ONE* **2017**, *12*, e0171410. [CrossRef]
21. Rahman, M.S.; Aktar, U.; Jani, M.R.; Shatabda, S. iPro70-FMWin: Identifying Sigma70 promoters using multiple windowing and minimal features. *Mol. Genet. Genom.* **2019**, *294*, 69–84. [CrossRef]
22. Shujaat, M.; Wahab, A.; Tayara, H.; Chong, K.T. pcPromoter-CNN: A CNN-Based Prediction and Classification of Promoters. *Genes* **2020**, *11*, 1529. doi: 10.3390/genes11121529. [CrossRef]
23. Zhang, M.; Li, F.; Marquez-Lago, T.T.; Leier, A.; Fan, C.; Kwoh, C.K.; Chou, K.C.; Song, J.; Jia, C. MULTiPly: A novel multi-layer predictor for discovering general and specific types of promoters. *Bioinformatics* **2019**, *35*, 2957–2965. [CrossRef]
24. Feng, P.; Ding, H.; Yang, H.; Chen, W.; Lin, H.; Chou, K.C. iRNA-PseColl: Identifying the occurrence sites of different RNA modifications by incorporating collective effects of nucleotides into PseKNC. *Mol. Ther.-Nucleic Acids* **2017**, *7*, 155–163. [CrossRef]

25. Chou, K.C. Some remarks on protein attribute prediction and pseudo amino acid composition. *J. Theory Biol.* **2011**, *273*, 236–247. [CrossRef]
26. Gama-Castro, S.; Salgado, H.; Santos-Zavaleta, A.; Ledezma-Tejeida, D.; Muñiz-Rascado, L.; García-Sotelo, J.S.; Alquicira-Hernández, K.; Martínez-Flores, I.; Pannier, L.; Castro-Mondragón, J.A.; et al. RegulonDB version 9.0: High-level integration of gene regulation, coexpression, motif clustering and beyond. *Nucleic Acids Res.* **2016**, *44*, D133–D143. [CrossRef]
27. Santos-Zavaleta, A.; Salgado, H.; Gama-Castro, S.; Sánchez-Pérez, M.; Gómez-Romero, L.; Ledezma-Tejeida, D.; García-Sotelo, J.S.; Alquicira-Hernández, K.; Muñiz-Rascado, L.J.; Peña-Loredo, P.; et al. RegulonDB v 10.5: Tackling challenges to unify classic and high throughput knowledge of gene regulation in *E. coli* K-12. *Nucleic Acids Res.* **2019**, *47*, D212–D220. [CrossRef]