Hi-Enhancer: a two-stage framework for prediction and localization of enhancers based on Blending-KAN and Stacking-Auto models

Aimin Li[1,†,*], Haotian Zhou[1,†], Rong Fei[1], Saurav Mallik[2], Xinhong Hei[1], Lei Wang[1]

[1]Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an University of Technology, Xi'an, 710048, Shaanxi, China.

[2]Department of Environmental Health, Harvard University T H Chan School of Public Health, Boston, 02115, MA, USA.

[*]Corresponding author: aimin.li@xaut.edu.cn; [†]Equal contribution.

Abstract

Motivation: Gene expression plays a crucial role in cell function, and enhancers can regulate gene expression precisely. Therefore, accurate prediction of enhancers is particularly critical. However, existing prediction methods have low accuracy or rely on fixed multiple epigenetic signals, which may not always be available.

Results: We proposed a two-stage framework that accurately predicts enhancers by flexibly combining multiple epigenetic signals. In the first stage, we designed a Blending-KAN model, which integrates the results of various base classifiers and employs Kolmogorov-Arnold Networks (KAN) as a meta-classifier to predict enhancers based on flexible combinations of multiple epigenetic signals. In the second stage, we developed a Stacking-Auto model, which extracted sequence features using DNABERT-2 and located the enhancers based on the Stacking strategy and AutoGluon framework. The accuracy of the Blending-KAN model reached 99.69% when five epigenetic signals were used. In cross-cell line prediction, the accuracy was more significant than or equal to 93.72%. With Gaussian noise, it still maintains an accuracy of 98.74%. In the second stage, the accuracy of the Stacking-Auto model is 80.50%, which is better than the existing 17 methods. The results show that our models can be flexibly used to predict and locate enhancers utilizing a combination of multiple epigenetic signals.

Availability and implementation: The source code for Hi-Enhancer is available via https://github.com/emanlee/Hi-Enhancer.

1 Introduction

The precise regulation of gene expression is essential for maintaining cellular function and organismal development. Regulatory elements such as transcription factors (TFs), promoters, and enhancers form this sophisticated regulatory network [1]. Enhancers are a special class of DNA sequences that can remotely regulate the transcription of genes and enhance or repress gene expression by interacting with promoters [2].

Although the role of enhancers in transcriptional regulation is widely recognized, it remains a challenge to predict enhancers accurately. Traditional methods for enhancer identification mainly rely on biological experimental techniques, which are usually time-consuming, costly, and complicated to operate [3]. In recent years, machine learning and deep learning-based methods have become essential tools for enhancer prediction. These methods can predict enhancers and their functional properties [4] from a large amount of

genomic data by integrating multiple data types, such as evolutionary conservation, epigenetic markers, DNA sequence motifs, and transcription factor binding sites. EnhancerFinder integrates DNA sequence motifs, evolutionary patterns, and functional genomic datasets of different cell types using a multinuclear learning approach to improve enhancer recognition [5]. iEnhancer-BERT is based on a pre-trained DNA language model and fine-tuned on the enhancer recognition task through a transfer learning strategy to extract deeper sequence features [6]. iEnhancer-DHF combines Pseudo k-Tuple Nucleotide Composition (PseKNC) and FastText methods for feature extraction and the Deep Neural Network (DNN) model for enhancer prediction [7]. DECODE utilizes five kinds of signals (DNase-seq as well as ChIP-seq data of H3K27ac, H3K4me3, H3K4me1, and H3K9ac), to train a deep neural network for accurately predicting cell type-specific enhancers [8]. In addition, DECODE implements a weakly supervised target detection framework for pinpointing enhancer boundaries.

Among all the above methods, DECODE has the highest accuracy in enhancer prediction. However, DECODE relies on five kinds of signals, which may not always be available.

This study aims to develop a new computational framework that can effectively predict and localize enhancers in a scenario where not all the above five kinds of signals are available. Similar to DECODE, our new framework, Hi-Enhancer, has two stages: i) predicting whether an enhancer exists within a genomic region and ii) if an enhancer exists, further localizing the boundaries of the enhancer.

In the first stage, we employed a model fusion technique to design a Blending-KAN model based on the AutoGluon framework [9] and the KAN model [10] for determining whether an enhancer is contained in a genomic region. Blending-KAN integrates 123 base classifiers through Blending [11]. We can efficiently identify the presence or absence of enhancers in genomic regions. Blending-KAN supports the flexible combination of multiple signals. It allows users to input one or more signals and obtains higher accuracy than DECODE.

In the second stage, we segmented the genomic regions containing enhancers and developed the Stacking-Auto model to localize the boundaries of enhancers. We used a sliding window to slice DNA sequences into 200 bp bins and extracted key features using DNABERT-2 [12]. Next, we designed the Stacking-Auto model to predict probabilities for each bin as an enhancer. Finally, based on these predicted probability values, we utilized a dynamic thresholding algorithm to pinpoint the complete boundaries of enhancers. The Stacking-Auto model combines the Stacking method [13] with the AutoGluon framework [9] to improve the model generalization performance, resulting in better performance than the existing 17 methods.

2 Materials and methods

2.1 Blending-KAN for predicting enhancer regions

2.1.1 Preprocessing datasets

We collected various data of the human cell lines HCT116 and A549 from the ENCODE data portal (https://www.encodeproject.org/). The data collected include STARR-seq, chromatin accessibility (DNase-seq), and ChIP-seq for H3K27ac, H3K4me3, H3K4me1, and H3K9ac [14]. We identified the overlap regions between the DNase-seq and STARR-seq peaks. If these overlapping regions intersect with any of the ChIP-seq peaks (H3K27ac, H3K4me3, H3K4me1, or H3K9aca), they are considered enhancer

regions (positive samples) [15]. Negative samples (non-enhancer regions) were randomly selected from other than enhancer regions. The length of both positive and negative samples is 4000 bp, and the number ratio is 1:10. Finally, we obtained 8350 positive and 83500 negative samples for model construction on the human cell line HCT116. For cross-cell line prediction, 3403 positive and 34030 negative samples were obtained on the human cell line HA549. DNase-seq and ChIP-seq signals were aggregated in each 10 bp bin and averaged. The length of the signal data for each sample was 400 after aggregation.

2.1.2 Blending-KAN model

Figure 1 illustrates the architecture of the Blending-KAN model, which uses Blending Learning [11]. Our model supports flexible combinations of kinds of signals (DNase-seq, H3K27ac, H3K4me3, H3K4me1, and H3K9ac). Base classifiers were trained via AutoGluon using the training set and then used to make predictions on the test set to generate a new set of prediction results. These predictions were input features for the KAN meta-classifier [10]. Our model is described in detail as follows.

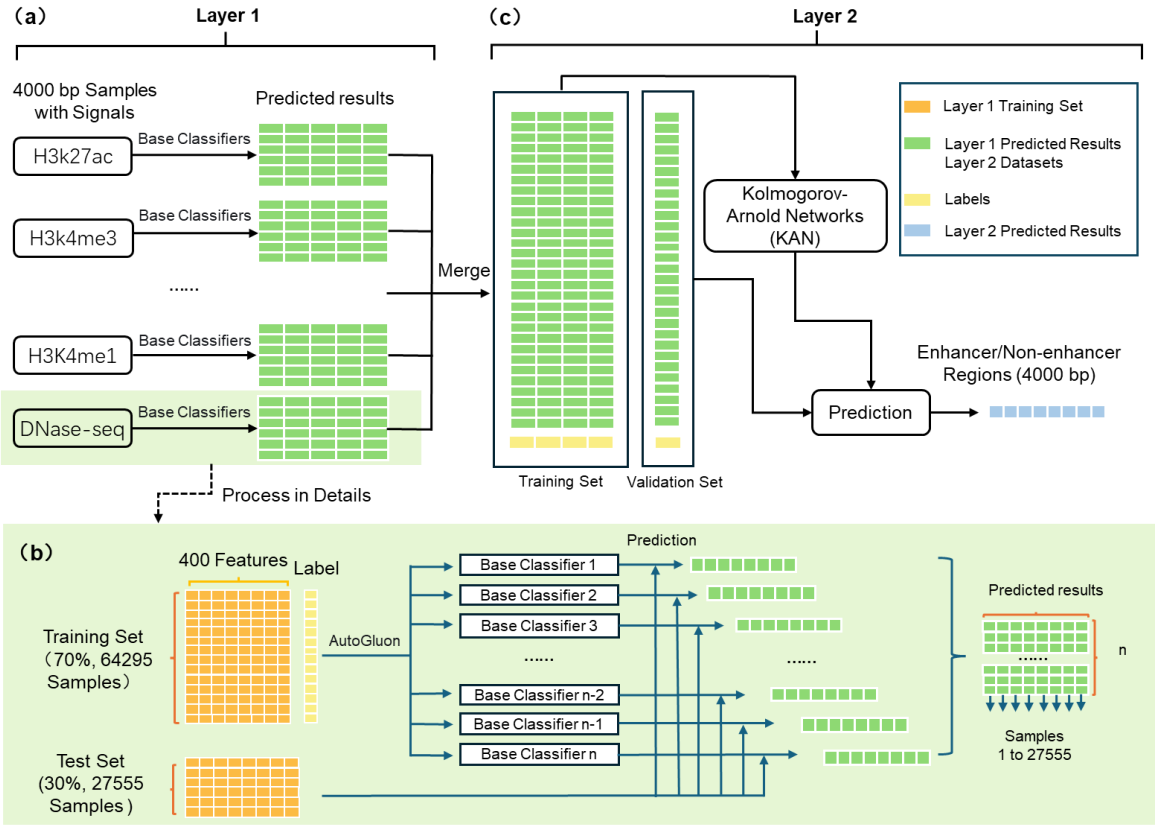

Figure 1. The architecture of the Blending-KAN model. (a) Layer 1 of Blending-KAN. (b) Details of layer 1. (c) Layer 2 of Blending-KAN.

(1) Layer 1 of Blending-KAN

The 4000 bp samples (enhancer and non-enhancer regions) were divided into training and test sets at 7:3 (see Supplementary Text S1). We used AutoGluon to train several models with different characteristics as components of the base classifiers. The prediction results of each base classifier on the test set were used as the input features of the second layer (Layer 2).

To address the class imbalance problem, we assigned weights to the positive and negative samples, with the weight ratio set to 10:1. Subsequently, the processed data were converted to AutoGluon's *TabularDataset* format, and the binary categorization model was

constructed using *TabularPredictor*. We used 5-fold bagging to enhance the robustness of the model and a preset parameter "*best_quality*" to optimize the model performance. We evaluated the test set and generated model performance rankings (see Supplementary Table S1).

The test set was predicted using AutoGluon's *TabularPredictor*. We used all models trained by AutoGluon as base classifiers. The prediction results of each base classifier on the test set were treated as input features to the meta-classifier (KAN) [10]. Precisely, we stacked the predictions of each base classifier by columns, which ultimately constituted a new feature set. This feature set contains the prediction information from different base classifiers and was eventually merged with the feature sets of other signal data as the input to the meta-classifier of the second layer (Layer 2).

(2) Layer 2 of Blending-KAN

We merged the results of Layer 1 into a new feature matrix for constructing the input of the second layer (Layer 2). In the training phase, KAN was used as the meta-classifier. A five-fold cross-validation strategy was adopted to ensure the reliability of the model evaluation. Stratified k-fold cross-validation was utilized to ensure that the class label distribution in each fold is consistent with the original data, thus avoiding the impact of class imbalance. During the training of each fold, the model was optimized by the Adam optimizer, and the cross-entropy loss function was used to train the network. The model updated the parameters by mini-batch gradient descent. In the validation phase, the model performed inference on the test set in no-gradient update mode and calculated the prediction accuracy. AUROC (Area Under the Receiver Operating Characteristic Curve) and AUPRC (Area Under the Precision-Recall Curve) were also evaluated to comprehensively reflect the model's classification performance and ensure the accuracy and robustness of the evaluation results.

2.2 Localization of the boundaries of enhancers

We designed the Stacking-Auto model to localize the boundaries of enhancers from the regions containing enhancers (Figure 2). First, we extracted the DNA sequences of the regions. Then, we split the sequences into 200 bp subsequences in a sliding window fashion (with a step size of 50 bp). Secondly, we designed the Stacking-Auto model to get the probability that subsequences were enhancers. Finally, we developed a dynamic thresholding algorithm to determine the boundaries of enhancers based on the probabilities.

2.2.1 Stacking-Auto model

To train a suitable model for determining the boundaries of enhancers, we used the benchmark datasets introduced in iEnhancer-2L [16] (see Supplementary Text S2 for details of benchmark datasets). We input the datasets into DNABERT-2 [12] to get embedded representations of samples (see Supplementary Text S3).

Stacking is an advanced integrated learning method that improves the generalization performance of a model by integrating the prediction results from different levels of learners. We used the LightGBM model [17] as the base learner for the first layer (Level 1). LightGBM becomes our first choice for the first-layer learner because of its ability to handle complex nonlinear relationships with high efficiency. We obtained the predictions of the first layer (Level 1) on the training set using ten-fold cross-validation. Then, we merged these predictions with the original input features to construct a new training set in Level 2. Subsequently, it was fed into a meta-learner to train a new final model.

While training the model in Level 1, we adopted the ten-fold cross-validation. The

predictions of each fold were summarized into a new set of features called "meta-features". After completing the ten-fold cross-validation, the meta-features of all the folds were combined into a complete feature set. These "meta-features" were stitched with the original features to form a richer and more comprehensive information representation. These expanded feature sets were used as inputs to the meta-learner. The meta-learner was also based on the AutoGluon framework, from which the best-performing model was selected as the meta-classifier. In the model training phase, we used AutoGluon's *TabularPredictor* class to build the model and train it using the *fit* method. After training, we used the *predict* method to predict the independent test set and comprehensively evaluated the model performance.

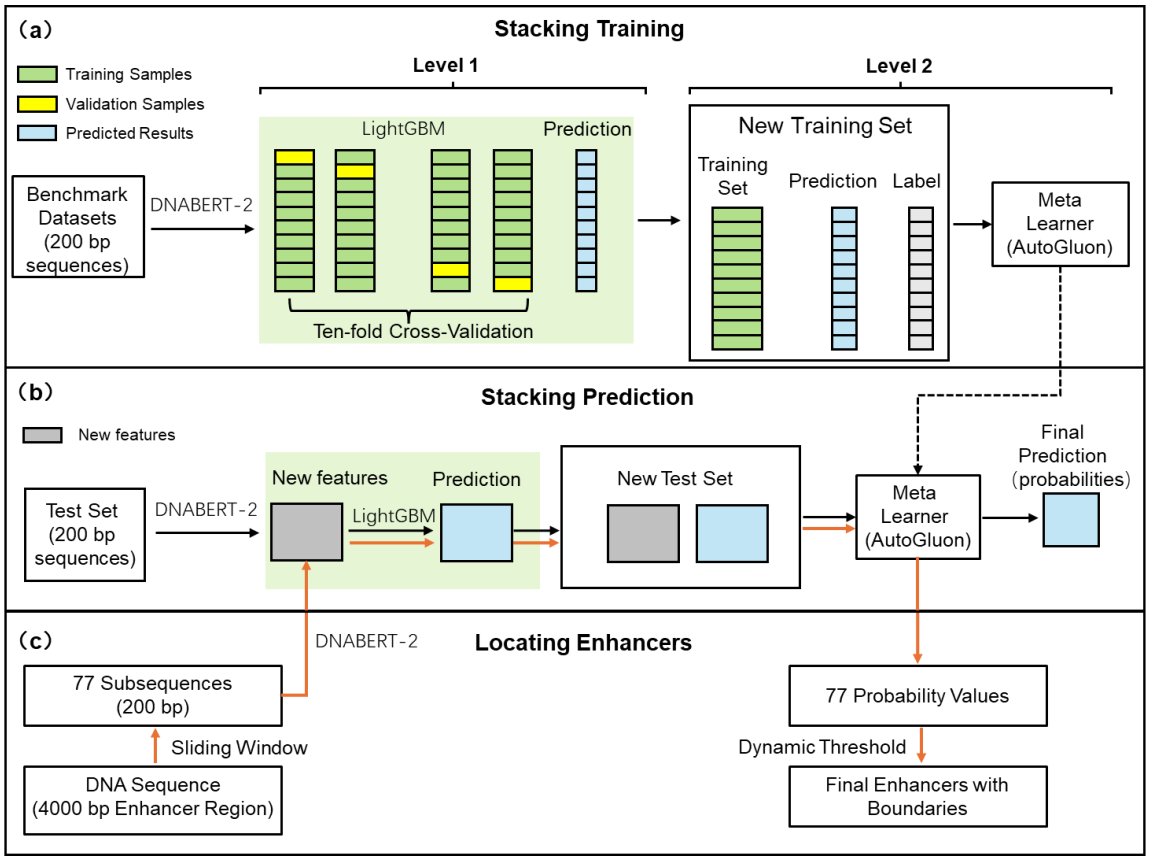

Figure 2. The architecture of the Stacking-Auto model. (a) Train the Stacking-Auto model. (b) Test the Stacking-Auto model. (c) Locating enhancers using the Stacking-Auto model.

2.2.2 Locating enhancers

To pinpoint the boundaries of an enhancer, we segmented a 4000 bp enhancer region into 77 subsequences using sliding windows (see Supplementary Text S4) and used DNABERT-2 to extract embedding features (Figure 2c). These embedding features were then processed by the Stacking-Auto model to generate the 77 probabilities of subsequences being enhancers. We designed a dynamic thresholding algorithm to pinpoint the boundaries of enhancers (see Supplementary Text S5).

3 Results

3.1 Performance of Blending-KAN on various signal combinations

Compared with DECODE, the Blending-KAN model has significant advantages, especially in supporting various epigenetic signal combinations. In this section, we show

the classification performance of Blending-KAN under different epigenetic signal combinations, including accuracy, AUROC, AUPRC, and running time (seconds) (Figure 3, see Supplementary Table S2 for details). These metrics reflect the effectiveness and superiority of Blending-KAN in handling various signal combinations.

(1) Manifestation of a single signal

The performance of Blending-KAN differed significantly when using one of five signals. Blending-KAN demonstrated optimal classification performance when using DNase-seq signals alone. Specifically, an accuracy of 0.9959 was achieved, with an AUROC of 0.9994 and an AUPRC of 0.9930. This result highlights the importance of chromatin accessibility in enhancer identification, demonstrating its strength in capturing sequence features. In contrast, single signals using histone modification (H3K27ac, H3K4me3, etc.) were slightly less impressive in performance, illustrating the limitations of single histone modification information in identifying complex biological functional regions. Run time for the single signal typically ranged from 4500 to 5200 seconds, suggesting that it is computationally efficient and particularly suitable for rapid initial screening.

(2) Advantages of the dual-signal combination

When using a two-signal combination, the performance of Blending-KAN is generally improved, especially in AUROC and AUPRC. For example, the combination of H3K4me1 with DNase-seq achieved an accuracy of 0.9956, with an AUROC of 0.9995 and an AUPRC of 0.9944. This result indicates that combining chromatin accessibility with histone modification signals can better capture the features of gene regulatory regions. Although the run time is about 9000 seconds, the performance improvement is significant, proving the advantage of multi-signal fusion.

(3) Boosting of three-signal combinations

When further increasing the number of signals to three, the overall performance of the model continued to improve, especially in terms of accuracy and AUPRC. For example, the combination of H3K4me3, H3K4me1, and DNase-seq achieved an accuracy of 0.9964, with an AUROC and AUPRC of 0.9997 and 0.9961, respectively. This suggests that an increase in the number of signal types helps the model to capture a richer set of features, which in turn improves the classification accuracy. The run time of the three-signal combination is approximately between 13500 and 14500 seconds, and its performance improvement is certainly worthwhile despite the increased computational cost.

(4) Balance of four-signal combinations

In the case of four-signal combinations, the overall performance of Blending-KAN is close to that of five-signal combinations. For example, the combination of H3K27ac, H3K9ac, H3K4me1, and DNase-seq achieves an accuracy of 0.9960, an AUROC of 0.9996, and an AUPRC of 0.9951. This combination achieves a good balance between the feature space and the model complexity. Still, at the same time, the performance of this combination is lower than the three-signal combination of H3K4me3, H3K4me1, and DNase-seq, which indicates that the signal combination should be flexibly selected according to the specific needs in practical applications.

(5) Optimal performance of five-signal combinations

Under the five-signal combination condition, Blending-KAN achieves the optimal classification performance. The accuracy of this combination is 0.9969, AUROC is 0.9997, and AUPRC is 0.9963, indicating that the combined use of the five signals can maximize

the feature information in the data and optimize the accuracy of enhancer detection. Although the run time grows to 23062 seconds, the enriched feature space allows Blending-KAN to optimize the performance in enhancer recognition.

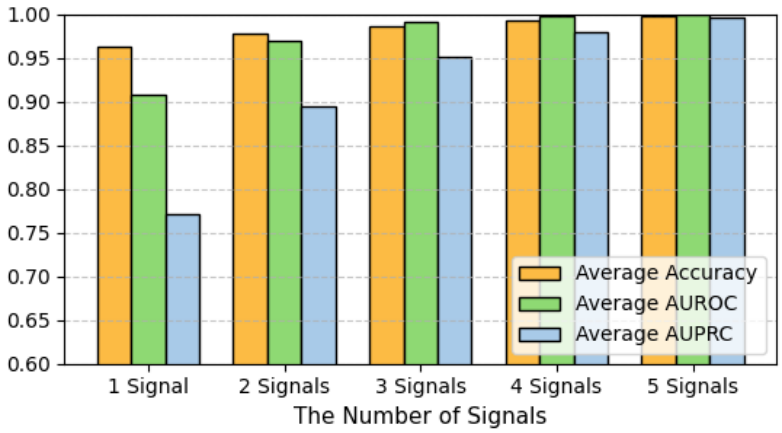

Figure 3. Classification performance of Blending-KAN under different combinations of epigenetic signals.

In summary, Blending-KAN performs excellent classification under various signal combinations, especially when combining DNase-seq and histone modification signals. By continuously increasing the variety of signals, Blending-KAN can effectively expand the feature space and thus improve the accuracy of enhancer detection. The flexibility of the model allows the flexible selection and combination of different signals to balance the cost of data acquisition and classification performance, providing an effective solution for the joint analysis of multiple signals. This characteristic has particular application in research environments where signal data acquisition is costly.

3.2 Robustness of the model to Gaussian noise

To assess the robustness of Blending-KAN in the face of noise, we added varying degrees of Gaussian noise to the data (H3k27ac, H3k4me3, H3k9ac, H3K4me1, and DNase-seq), with the noise standard deviation (σ) ranging from 0.1 to 0.9, and the results were cross-validated in a five-fold (see Supplementary Text S6).

As shown in Table 1, the classification performance of Blending-KAN shows a certain decreasing trend after adding different levels of Gaussian noise in signals, especially in the high noise condition.

Table 1. Blending-KAN's performance on data with Gaussian noise.

Noise (σ)	0.1	0.3	0.5	0.7	0.9
Accuracy	0.9956	0.9929	0.9934	0.9875	0.9874
AUROC	0.9992	0.9983	0.9983	0.9952	0.9954
AUPRC	0.9987	0.9784	0.9757	0.9631	0.9635

Low noise level ($\sigma = 0.1$): the model performs best when the noise standard deviation is 0.1, with an accuracy of 0.9956, an AUROC of 0.9992, and an AUPRC of 0.9987. This indicates that Blending-KAN can stably classify augmented subregions at lower noise levels, maintaining high accuracy and discriminative power.

Medium noise level ($\sigma = 0.3$ and $\sigma = 0.5$): the performance of Blending-KAN decreases at noise standard deviations of 0.3 and 0.5. The accuracies are 0.9929 and 0.9934,

respectively, and the AUROC and AUPRC also show a slight decrease. This suggests that the moderate noise level has some impact on the classification accuracy, especially in terms of AUPRC, where the model performance is degraded.

High noise level ($\sigma = 0.7$ and $\sigma = 0.9$): the performance of Blending-KAN decreases significantly as the noise standard deviation increases to 0.7 and 0.9. The accuracy drops to 0.9875 and 0.9874, the AUROC to 0.9952 and 0.9954, and the AUPRC to 0.9631 and 0.9635, respectively. Nevertheless, the model still maintains relatively high accuracy, indicating that Blending-KAN is still able to perform enhancer identification effectively in noisy situations but has limited ability for higher noise adaptation ability is limited.

By adding different levels of Gaussian noise and performing five-fold cross-validation, the experimental results show that the Blending-KAN model can cope better with low to moderate noise levels and still maintain high classification performance. However, as the standard deviation of the noise increases, the model performance decreases accordingly, especially in the AUPRC metric. The significant decrease in AUPRC may indicate that the increase in noise reduces the model's precision and recall in distinguishing between different classes.

3.3 Performance of Blending-KAN on cross-cell line prediction

To evaluate the performance of Blending-KAN in cross-cell line prediction, we trained the model using data from the HCT116 cell line and performed forecasts with data from the A549 cell line (Figure 4).

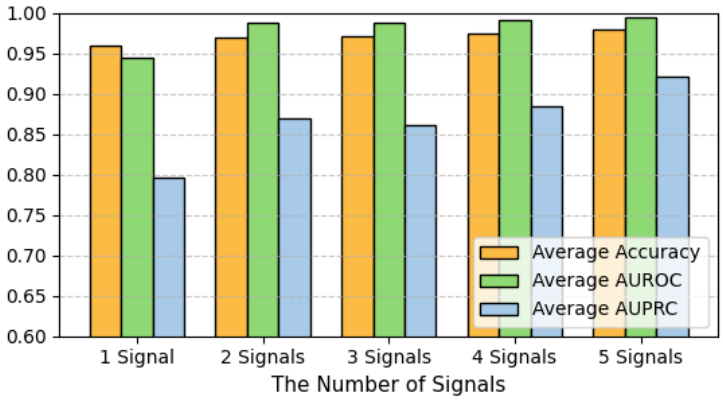

Figure 4. Performance of Blending-KAN on cross-cell line prediction.

(1) Cross-cell line prediction using a kind of signal

Supplementary Table S3 demonstrates the results of cross-cell line prediction using a kind of signal. The performance varied widely with a single signal for cross-cell line prediction. When using H3K27ac, the accuracy was 0.9566, AUROC was 0.9748, AUPRC was 0.8358, and the run time was 848 seconds. When using other single signals, the accuracies fluctuated between 0.9372 and 0.9700, suggesting that different epigenetic signals contribute differently to the transcellular lineage prediction.

(2) Cross-cell line prediction using two kinds of signals

The cross-cell line prediction performance of Blending-KAN was generally significantly improved on signal combinations. Supplementary Table S4 demonstrates the cross-cell line prediction using two types of signals. When combining H3K27ac and H3K4me3, the accuracy reached 0.9676, the AUROC was 0.9830, and the AUPRC was 0.8587.

(3) Cross-cell line prediction using three or more kinds of signals

The prediction performance of Blending-KAN is further improved as the number of signal combinations increases. Supplementary Table S5 demonstrates the results of cross-cell line prediction with multiple signal combinations, which shows a very high prediction accuracy of 0.9688, AUROC of 0.9924, and AUPRC of 0.9147 when trained with three signals (H3k27ac+H3k4me3+DNase-seq). When using a combination of four signals (H3k27ac+H3k4me3+H3k9ac+DNase-seq), the model achieved an accuracy of 0.9676, an AUROC of 0.9818, and an AUPRC of 0.7546, which is a slight decrease in performance but still better than a single signal.

(4) Best performance of five kinds of signal combinations

Among all signal combinations, Blending-KAN trained with five signals performed the best, with an accuracy of 0.9798, an AUROC of 0.9942, and an AUPRC of 0.9209. Although the run time increased compared to the single signal and dual-signal combinations, the model's performance in cross-cell line prediction was optimal thanks to the rich features provided by the five signals.

Experimental results of Blending-KAN in cross-cell line prediction show that the classification performance of the model is significantly improved with the increase in the number of signals. In particular, under the combination of five epigenetic signals, Blending-KAN can effectively fuse multiple signal information, improving the accuracy and stability of enhancer recognition.

3.4 Comparison of Stacking-Auto with existing methods

We also proposed a Stacking-Auto model that uses DNABERT-2 to extract features and AutoGluon as a base classifier and meta-classifier aiming at locating enhancers. To evaluate the effectiveness of Stacking-Auto, we compared and analyzed it with 17 existing methods, including iEnhancer-2L [16], EnhancerPred [18], iEnhancer-DSNet [19], iEnhancer-Deep [20], etc. The metrics are accuracy, sensitivity, specificity, and Matthews's correlation coefficient (MCC). The results are shown in Table 2.

Stacking-Auto performs well in several metrics, especially regarding accuracy and MCC. Specifically, the accuracy of Stacking-Auto reaches 80.50%, which is slightly higher than the 79.75% of iEnhancer-MRBF [21]. Stacking-Auto's sensitivity is 80.50%, which maintains a high accuracy while considering the higher specificity of 80.50%. 80.50% is the highest among all methods, showing a strong ability in negative sample recognition. Notably, Stacking-Auto achieves an MCC value of 0.61, the highest among all methods. This indicates that the model is better balanced in dealing with the class imbalance problem and can locate the augmented subsequence more stably.

Our Stacking-Auto model outperforms existing methods, mainly due to its significant advantages in feature extraction, automated learning, multi-model integration, and meta-classifier optimization. First, with DNABERT-2, we can extract deep DNA sequence features, which provides richer biological information for prediction and improves data quality. Meanwhile, the automation feature of AutoGluon allows the model to automatically complete feature engineering, model selection, and hyper-parameter tuning, significantly reducing human intervention and improving generalization ability. In addition, Stacking-Auto employs the Stacking strategy to integrate the prediction results of base classifiers and enhance the overall prediction performance through meta-classifier optimization integration. Selected meta-classifiers further optimize these integration results, enabling Stacking-Auto to demonstrate higher accuracy and stability in the augmented subsequence localization task.

Table 2. Performance comparison of Stacking-Auto with 17 existing methods.

Methods	Accuracy	Sensitivity	Specificity	MCC
iEnhancer-2L[16]	73.00	71.00	75.00	0.46
EnhancerPred[18]	74.00	73.50	74.50	0.48
iEnhancer-DSNet[19]	78.00	78.00	77.00	0.56
iEnhancer-Deep[20]	74.02	81.5	67.00	0.49
iEnhancer-EL[22]	74.45	71.00	78.50	0.50
Rank-GAN[23]	75.25	74.87	75.63	0.51
Le et al.'s BERT[24]	75.60	80.00	71.20	0.51
iEnhancer-XG[25]	75.75	74.00	77.50	0.51
Tan et al. Enhancer[26]	76.00	76.00	76.00	0.51
iEnhancer-ECNN[27]	76.90	78.50	75.20	0.54
iEnhancer-EBLSTM [28]	77.20	75.50	79.50	0.53
iEnhancer-CNN[29]	77.50	78.25	79.00	0.59
iEnhancer-DCLA[30]	78.25	78.00	78.50	0.57
iEnhancer-GAN[31]	78.40	81.10	75.80	0.57
Enhancer-RD[32]	78.80	81.00	76.50	0.58
iEnhancer-5Step[33]	79.00	**82.00**	76.00	0.58
iEnhancer-MRBF[21]	79.75	**82.00**	77.50	0.60
Ours	**80.50**	80.50	**80.50**	**0.61**

4. Discussion

This study proposes a new framework for enhancer prediction and localization based on Blending-KAN and Stacking-Auto models. First, enhancer and non-enhancer regions are predicted on multidimensional epigenetic signals using Blending-KAN. Enhancer regions are segmented into 200 bp subsequences by a sliding window, and Stacking-Auto further predicts the probability of each subsequence being an enhancer. Ultimately, based on these probability values, a dynamic thresholding algorithm is used to determine the boundaries of enhancers accurately. The accuracy of our framework outperforms existing methods.

4.1 The advantages of Blending-KAN in enhancer prediction

The Blending-KAN model efficiently predicts enhancer regions by integrating different epigenetic signals, combining the weighted fusion of multi-base classifiers and the nonlinear mapping advantage of KAN. Experimental results show that introducing multiple signal combinations (e.g., DNase-seq and H3K4me1) significantly improves the classification accuracy and robustness. KAN can further explore the nonlinear relationship between different signals, thus enhancing the model's ability to recognize complex regulatory regions. In addition, the weighted fusion method performs well in coping with the class imbalance problem, which makes Blending-KAN have stable prediction effects under different noise levels.

4.2 A novel strategy for enhancer localization

For enhancer localization, we used a sliding window strategy to partition a 4000 bp sequence into 200 bp overlapping subsequences and calculated the probability of each subsequence being an enhancer using Stacking-Auto. Stacking-Auto effectively captures sequence features in different subsequences through DNABERT-2 feature extraction, AutoGluon, and Stacking. Through the dynamic thresholding algorithm of probability

mean and standard deviation, we flexibly extracted the high-probability regions and merged the neighboring segments to determine the complete enhancers. This method is highly adaptable to different samples and effectively improves the accuracy of enhancer localization.

Funding: This work is supported by the National Natural Science Foundation International cooperation and exchange projects (62120106011), the National Natural Science Foundation of China (62176146, U2468206), and the Natural Science Basic Research Program of Shaanxi (2024JC-YBMS-484).

Conflict of Interest: none declared.

References

1. Cramer P: **Organization and regulation of gene transcription**. *Nature* 2019, **573**(7772):45-54.
2. Dao LT, Spicuglia S: **Transcriptional regulation by promoters with enhancer function**. *Transcription* 2018, **9**(5):307-314.
3. Boyle AP, Araya CL, Brdlik C *et al*: **Comparative analysis of regulatory information and circuits across distant species**. *Nature* 2014, **512**(7515):453-456.
4. Ernst J, Kellis M: **ChromHMM: automating chromatin-state discovery and characterization**. *Nat Methods* 2012, **9**(3):215-216.
5. Erwin GD, Oksenberg N, Truty RM *et al*: **Integrating diverse datasets improves developmental enhancer prediction**. *PLoS computational biology* 2014, **10**(6):e1003677.
6. Luo H, Chen C, Shan W *et al*: **iEnhancer-BERT: a novel transfer learning architecture based on DNA-language model for identifying enhancers and their strength**. In: *International Conference on Intelligent Computing: 2022*: Springer; 2022: 153-165.
7. Inayat N, Khan M, Iqbal N *et al*: **iEnhancer-DHF: identification of enhancers and their strengths using optimize deep neural network with multiple features extraction methods**. *Ieee Access* 2021, **9**:40783-40796.
8. Chen Z, Zhang J, Liu J *et al*: **DECODE: A De ep-learning Framework for Co n de nsing Enhancers and Refining Boundaries with Large-scale Functional Assays**. *Bioinformatics* 2021, **37**(Supplement_1):i280-i288.
9. Erickson N, Mueller J, Shirkov A *et al*: **Autogluon-tabular: Robust and accurate automl for structured data**. *arXiv preprint arXiv:200306505* 2020.
10. Liu Z, Wang Y, Vaidya S *et al*: **Kan: Kolmogorov-arnold networks**. *arXiv preprint arXiv:240419756* 2024.
11. Breiman L: **Stacked regressions**. *Machine learning* 1996, **24**:49-64.
12. Zhou Z, Ji Y, Li W *et al*: **Dnabert-2: Efficient foundation model and benchmark for multi-species genome**. *arXiv preprint arXiv:230615006* 2023.
13. Wolpert DH: **Stacked generalization**. *Neural networks* 1992, **5**(2):241-259.
14. Feingold E, Good P, Guyer M *et al*: **The ENCODE (ENCyclopedia of DNA elements) project**. *Science* 2004, **306**(5696):636-640.
15. Zhang Y, Liu T, Meyer CA *et al*: **Model-based analysis of ChIP-Seq (MACS)**. *Genome biology* 2008, **9**:1-9.
16. Liu B, Fang L, Long R *et al*: **iEnhancer-2L: a two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition**. *Bioinformatics* 2016, **32**(3):362-369.
17. Li H, Xu Z, Taylor G *et al*: **Visualizing the loss landscape of neural nets**. *Advances in neural information processing systems* 2018, **31**.
18. Jia C, He W: **EnhancerPred: a predictor for discovering enhancers based on the combination and selection of multiple features**. *Scientific reports* 2016, **6**(1):38741.

19. Asim MN, Ibrahim MA, Malik MI *et al*: **Enhancer-dsnet: a supervisedly prepared enriched sequence representation for the identification of enhancers and their strength**. In: *Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 23–27, 2020, Proceedings, Part III 27: 2020*: Springer; 2020: 38-48.
20. Kamran H, Tahir M, Tayara H *et al*: **iEnhancer-deep: a computational predictor for enhancer sites and their strength using deep learning**. *Applied Sciences* 2022, **12**(4):2120.
21. Xiao Z, Wang L, Ding Y *et al*: **iEnhancer-MRBF: Identifying enhancers and their strength with a multiple Laplacian-regularized radial basis function network**. *Methods* 2022, **208**:1-8.
22. Liu B, Li K, Huang D-S *et al*: **iEnhancer-EL: identifying enhancers and their strength with ensemble learning approach**. *Bioinformatics* 2018, **34**(22):3835-3842.
23. Geng Q, Yang R, Zhang L: **A deep learning framework for enhancer prediction using word embedding and sequence generation**. *Biophysical Chemistry* 2022, **286**:106822.
24. Le NQK, Ho Q-T, Nguyen T-T-D *et al*: **A transformer architecture based on BERT and 2D convolutional neural network to identify DNA enhancers from sequence information**. *Briefings in bioinformatics* 2021, **22**(5):bbab005.
25. Cai L, Ren X, Fu X *et al*: **iEnhancer-XG: interpretable sequence-based enhancers and their strength predictor**. *Bioinformatics* 2021, **37**(8):1060-1067.
26. Tan KK, Le NQK, Yeh H-Y *et al*: **Ensemble of deep recurrent neural networks for identifying enhancers via dinucleotide physicochemical properties**. *Cells* 2019, **8**(7):767.
27. Nguyen QH, Nguyen-Vo T-H, Le NQK *et al*: **iEnhancer-ECNN: identifying enhancers and their strength using ensembles of convolutional neural networks**. *BMC genomics* 2019, **20**:1-10.
28. Niu K, Luo X, Zhang S *et al*: **iEnhancer-EBLSTM: identifying enhancers and strengths by ensembles of bidirectional long short-term memory**. *Frontiers in Genetics* 2021, **12**:665498.
29. Khanal J, Tayara H, Chong KT: **Identifying enhancers and their strength by the integration of word embedding and convolution neural network**. *Ieee Access* 2020, **8**:58369-58376.
30. Liao M, Zhao J-p, Tian J *et al*: **iEnhancer-DCLA: using the original sequence to identify enhancers and their strength based on a deep learning framework**. *BMC bioinformatics* 2022, **23**(1):480.
31. Yang R, Wu F, Zhang C *et al*: **iEnhancer-GAN: a deep learning framework in combination with word embedding and sequence generative adversarial net to identify enhancers and their strength**. *International Journal of Molecular Sciences* 2021, **22**(7):3589.
32. Yang H, Wang S, Xia X: **iEnhancer-RD: identification of enhancers and their strength using RKPK features and deep neural networks**. *Analytical Biochemistry* 2021, **630**:114318.
33. Le NQK, Yapp EKY, Ho Q-T *et al*: **iEnhancer-5Step: identifying enhancers using hidden information of DNA sequences via Chou's 5-step rule and word embedding**. *Analytical biochemistry* 2019, **571**:53-61.