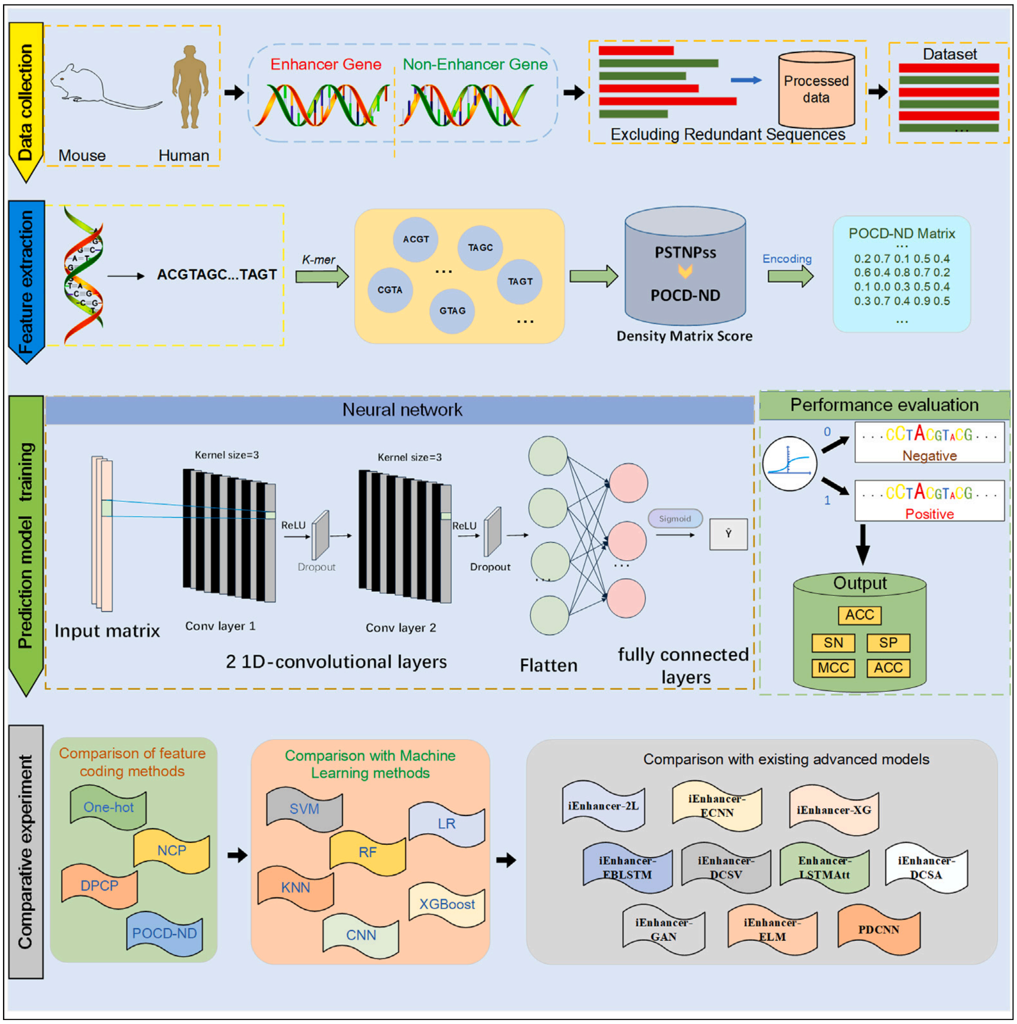
iScience

CellPress
OPEN ACCESS

Article

A deep learning model for DNA enhancer prediction based on nucleotide position aware feature encoding



Wenxing Hu, Yelin Li, Yan Wu, Lixin Guan, Mengshan Li

msli@gnnu.edu.cn

Highlights

A deep learning model PDCNN for identifying DNA enhancers

Improving model performance with position-aware encoders

Comparative studies have shown that PDCNN is superior to existing models

Hu et al., iScience 27, 110030
June 21, 2024 © 2024 The Author(s). Published by Elsevier Inc.
https://doi.org/10.1016/j.isci.2024.110030

iScience

Article

A deep learning model for DNA enhancer prediction based on nucleotide position aware feature encoding

Wenxing Hu,[1] Yelin Li,[1] Yan Wu,[1] Lixin Guan,[1] and Mengshan Li[1,2,*]

SUMMARY

Enhancers, genomic DNA elements, regulate neighboring gene expression crucial for biological processes like cell differentiation and stress response. However, current machine learning methods for predicting DNA enhancers often underutilize hidden features in gene sequences, limiting model accuracy. Hence, this article proposes the PDCNN model, a deep learning-based enhancer prediction method. PDCNN extracts statistical nucleotide representations from gene sequences, discerning positional distribution information of nucleotides in modifier-like DNA sequences. With a convolutional neural network structure, PDCNN employs dual convolutional and fully connected layers. The cross-entropy loss function iteratively updates using a gradient descent algorithm, enhancing prediction accuracy. Model parameters are fine-tuned to select optimal combinations for training, achieving over 95% accuracy. Comparative analysis with traditional methods and existing models demonstrates PDCNN's robust feature extraction capability. It outperforms advanced machine learning methods in identifying DNA enhancers, presenting an effective method with broad implications for genomics, biology, and medical research.

INTRODUCTION

Enhancers, regulatory non-coding DNA fragments, bind to specific transcription factors, amplifying the transcription process of relevant genes and playing a pivotal role in gene expression regulation.[1,2]They exhibit diverse functional subgroups, including accumulating enhancers, latent enhancers, strong enhancers, and weak enhancers.[3] These interact with transcription factors to regulate the transcription of target genes by attracting elongation factors or initiating RNA polymerase II,[4,5] as depicted in Figure 1.

As research has advanced, associations have emerged between genetic variations in enhancers and various human diseases, such as different types of inflammatory bowel disease and cancer.[6–8] This realization has spurred an urgent need for more comprehensive research and understanding of enhancers. The identification and classification of enhancers have become prominent research topics in bioinformatics and computational biology. However, the dynamic nature of enhancers, which may be distributed up to one trillion base pairs away from the target gene and exist across multiple chromosomes,[9] introduces new challenges to their identification and classification.

Given the functional significance of enhancers in promoting gene expression, pinpointing their locations in the genome constitutes a focal point for laboratory researchers and computational biologists. Historically, enhancer prediction heavily relied on biological experimental techniques. For instance, conserved analyses utilized sequence conservation data and transcription factor binding site data for prediction.[6,10,11] However, this approach has limitations, as transcription factors do not consistently occupy all enhancer sites, and their associated targets may also be repressed. Recent advancements in next-generation sequencing (NGS) have substantially eased the assessment of functional enhancer activity.[12,13] Nevertheless, these experimental methods are resource-intensive and time-consuming, applicable to only a limited number of cell types. Current research trends lean toward the development of computational methods. Various machine learning approaches have been explored for recognizing DNA enhancer sequences.[14]

In the realm of predicting enhancers in the human genome, various machine learning (ML) computational methods have been proposed, such as ChromaGenSVM,[15] CSI-ANN,[16] RFECS,[17] EnhancerFinder,[18] Sbper[19] and BiRen.[20] iEnhancer-2L[21] is the first predictive model introduced by Liu et al. that can identify both enhancers and their respective strengths. It employs the pseudo k-tuple nucleotide composition (PseKNC) as the encoding method for sequence features. Subsequently, Liu et al., building upon the support vector machine (SVM), utilized the pseudo degenerate k-mer nucleotide composition (PseDekNC) to extract features from DNA sequences, leading to the development of the iEnhancer-PsedekNC predictor.[22] EnhancerPred is based on Liu's dataset and employs bilateral Bayesian and pseudo nucleotide composition as feature extraction methods. It constructs a two-layer predictor through wrapper-based two-step feature selection.[23] Nguyen et al., using One-hot encoding and k-mers as input, established an ensemble framework based on Convolutional Neural Network (CNN) known as

[1]College of Physics and Electronic Information, Gannan Normal University, Ganzhou 341000, Jiangxi, China
[2]Lead contact
*Correspondence: msli@gnnu.edu.cn

https://doi.org/10.1016/j.isci.2024.110030

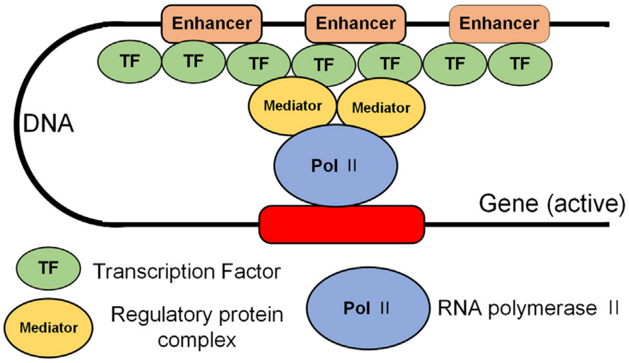


Figure 1. The structural characterization and function of enhancers

iEnhancer-ECNN.[24] Cai et al., by combining five feature sets, including k-spectrum profiles, mismatch k-tuples, Position-Specific Scoring Matrix (PSSM), and Pseudo-Dinucleotide Composition (PseDNC), utilized "XGBoost" as the base classifier to build a two-layer predictor named iEnhancer-XG.[25] iEnhancer-EBLSTM encodes the input DNA sequence with 3-mers and predicts enhancers through bidirectional LSTM.[26] In machine learning approaches to predict enhancers, the extraction of features from sequences is crucial. It determines whether valuable feature information can be mined from genomic data for model learning. To capture more features from DNA enhancer sequences, Jia et al. combined One-hot encoding and NCP encoding. They employed an enhanced DenseNet and improved CBAM attention modules for prediction, referring to this model as iEnhancer-DCSV.[27] Wang et al. chose to tokenize the sequence initially using n-gram and skip-gram, followed by One-hot encoding. They eventually developed the iEnhancer-DCSA model, a convolutional neural network employing dual-scale fusion.[28] Basith et al. utilized seven encodings, including DPCP and k-mer, and integrated five machine learning methods, such as RF, SVM, and XGB, to establish an enhancer prediction model.[29] Le et al. employed bidirectional encoder representations from transformers (BERT) in conjunction with a convolutional neural network for enhancer prediction.[30] Yang et al. introduced iEnhancer-GAN, a model that combines word embedding skip-gram to convert words into vectors, utilizing a convolutional neural network architecture for recognition tasks.[31] Alakuş developed an encoding method for feature learning based on sequence frequency, utilizing three DNA encoding schemes: EIIP, integer, and atomic number.[32] Additionally, there are efforts to optimize machine learning models to enhance recognition accuracy. For example, Huang et al. employed a simple encoding of sequences, utilizing bidirectional long short-term memory and attention block-based deep learning methods to develop the Enhancer-LSTMAtt predictor.[33] Kuar et al. leveraged DNA structural features, combining natural language processing, convolutional neural networks, and long short-term memory to accurately predict enhancers in genomic data, a model referred to as PEDH.[34] Furthermore, enhancer recognition methods, such as iEnhancer-ELM based on a BERT-like enhancer language model (DNABERT),[35] and iEnhancer-BERT,[36] a transfer learning approach based on pre-trained DNA language models, have also been introduced. In summary, there is a growing body of research exploring the application of machine learning methods in DNA enhancer prediction, demonstrating promising performance and significant progress.[37–42] However, existing methods have limitations in terms of prediction effectiveness and generalizability. This is primarily due to the use of simple numerical sequence encoding methods that fail to capture the position-specific distributional information of nucleotides. Effectively characterizing constant and discriminative regions of nucleotides in DNA sequences is crucial for accurately predicting DNA enhancer categories.

To comprehensively exploit the features within DNA gene sequences and uncover hidden information in nucleotides for accurate prediction of the crucial impact of DNA enhancers, the choice of a feature encoding method with robust representation capabilities is paramount. In this context, the primary contribution of this research lies in the application of a DNA sequence encoding method known as the Position-Specific Nucleotide Density-based generalized encoding (POCD-ND). Compared to traditional numerical representations and other common encoding methods, the POCD-ND encoder focuses on capturing the distribution information of nucleotides at different positions within DNA sequences, introducing elements aware of position. This enables the proposed model to more accurately characterize the conserved and discriminative regions of nucleotides within DNA sequences, thus exhibiting superior performance in predicting DNA enhancers. This unique encoding method not only successfully tackles the complexity of the original sequences but more importantly, provides an innovative approach for the application of deep learning models, laying the groundwork for subsequent research. To maximize the utilization of the distinctive features of the POCD-ND encoder and lay a foundation for the application of subsequent deep learning models, a Convolutional Neural Network model based on this encoding (PDCNN) was introduced(detailed in Figure 2). Unlike conventional DNA sequence processing methods, the PDCNN model efficiently captures the distribution patterns of nucleotide groups at different positions within the convolutional layers, allowing the model to sensitively capture key features within DNA enhancer sequences. This integrated architecture endows our model with a unique advantage in DNA enhancer prediction tasks. To establish a more generalizable model, we autonomously constructed a dataset for training the PDCNN model and experimentally determined the optimal sequence length for the training dataset. To validate the model's effectiveness, the trained PDCNN model was compared with several commonly used encoding methods and classical machine learning models. By fine-tuning the hyperparameters of the PDCNN model, the parameter combination yielding the optimal predictive performance was selected for training. Utilizing gradient descent algorithms, we iteratively optimized the cross-entropy loss function to better achieve

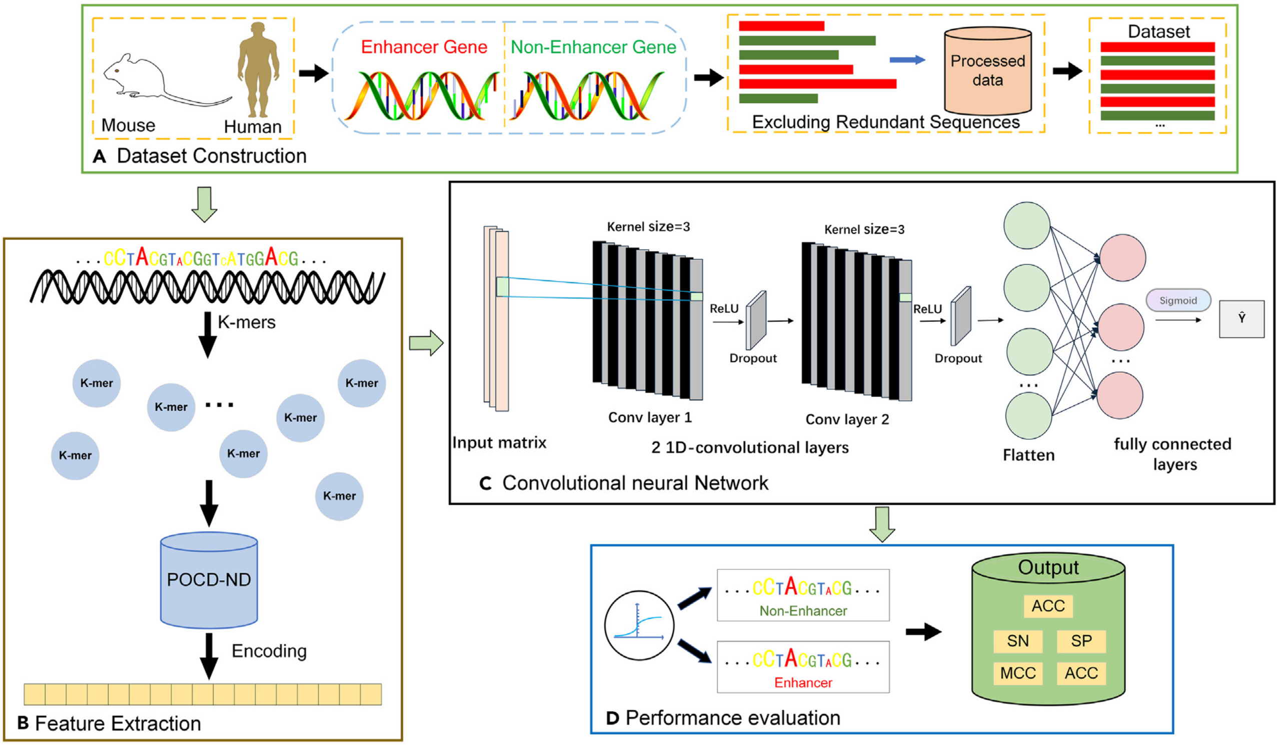


Figure 2. Workflow of model PDCNN to predict DNA enhancers

(A) Data Construction.

(B) Feature Extraction.

(C) Convolutional neural Network.

(D) Performance evaluation.

identification and classification of DNA enhancer tasks. These series of innovative steps not only theoretically strengthen our proposed model but also significantly improve performance in experimental results, bringing new insights and practicality to the field of deep learning-based DNA enhancer prediction.

RESULTS AND DISCUSSION

Effect of different sequence lengths and k-mers on model performance

Due to the variable sequence lengths of the downloaded human and mouse DNA enhancer data from the VISTA Enhancer Browser, and an approximate 1:1 ratio between positive and negative samples of enhancers, as shown in Table S1. The human gene sequences range from a minimum length of 453bp to a maximum of 11052bp, with similar patterns observed in the mouse data. To determine the optimal sequence length for the model's prediction of DNA enhancers, the sequences from the original dataset were divided into six lengths: 50bp, 100bp, 150bp, 200bp, 250bp, and 300bp, ensuring equal sequence lengths for the data. At the same time, we chose to adjust the ratio of positive to negative samples to approximately 1:3. From the partitioned positive samples, a random selection of one-third of the negative samples was taken as experimental data, aiming to reflect the actual distribution in the biological and medical fields. In various biological and medical applications, positive samples are relatively scarce compared to an abundance of negative samples, resulting in a noticeable imbalance in their distribution. This imbalance mirrors common scenarios in biological experiments and practical applications. Therefore, our purpose in adjusting the sample ratio is to more accurately simulate the data distribution in the real world. After partitioning the dataset into different lengths, the corresponding increase in data volume enhances the dataset's capacity to train the model, consequently improving the model's performance. Table 1 provides information on the divided data.

The choice of k-mer size directly impacts the feature representation of DNA enhancer sequences and consequently influences the model's prediction performance through the generation of distinct POCD-ND score matrices. 1-mer and 2-mer features exhibit higher frequencies in sequences, with approximately the same frequency across different categories, classifying them as frequent features. However, their distributions in sequences significantly differ. On the other hand, 3-mer, 4-mer, and 5-mer features vary considerably with increasing k and are less frequent. This experiment aims to analyze whether different k-mers yield better features for the model. Models were constructed based on data with varying sequence lengths, and different k-mers were combined separately. To ensure models with strong generalization ability, hyperparameters

Table 1. Detail information of the different sequence lengths in the datasets

		Training dataset		Testing dataset	
Dataset	Sequence length	Positive	Negative	Positive	Negative
Human	50bp	9370	28112	1017	3061
	100bp	4745	14236	534	1603
	150bp	3142	9426	332	998
	200bp	2318	6954	255	767
	250bp	1804	5413	202	608
	300bp	1508	4526	167	501
Mouse	50bp	12685	38056	1405	4217
	100bp	6241	18724	669	2009
	150bp	4079	12239	452	1356
	200bp	3063	9189	340	1021
	250bp	2395	7186	264	793
	300bp	1994	5982	219	657

were optimized, and models were cross-validated using 10-fold cross-validation. The evaluation results of the trained models are illustrated in Figure 3.

For the human dataset, in Figure 3A, it is noticeable that the SN values for the 6 sequence lengths and the 1-mer combination are generally low. Only at a sequence length of 250bp does it surpass 80%. However, in Figure 3B, the SP values are notably low. This indicates that for the 1-mer, the model performs well in identifying DNA enhancers at a sequence length of 250bp but struggles to correctly identify non-enhancer data, leading to an overall inadequate recognition performance. From the evaluation of the model's ACC values in Figure 3G, it is evident that the ACC values for 1-mer are lower than for other k-mers. This suggests that the POCD-ND score matrix generated by 1-mer fails to distinctly represent the features of both enhancers and non-enhancers, thereby resulting in the model's unsatisfactory ability to predict enhancers. SN values represent the model's ability to recognize positives, while SP values represent the model's ability to recognize negatives. Among the values depicted in the figure, 3-mer, 4-mer, and 5-mer all exhibit SN and SP values exceeding 80%. Among these k-mers, as observed in Figure 3G, the model achieves the highest ACC values at a sequence length of 200bp, reaching 94.4%, 93.7%, and 92.4%, respectively, surpassing ACC values for other sequence lengths. This suggests that the optimal sequence length for predicting DNA enhancers by the model is 200bp, consistent with the sequence length of the DNA enhancer data used by Liu et al. Figures 3C and 3F illustrate the evaluation results of MCC and AUC values for different k-mers of gene sequences with a length of 200bp. Notably, the MCC and AUC values for 3-mer are higher than for other k-mers, indicating that the POCD-ND score matrix generated by 3-mer provides a superior feature representation, thus enhancing the predictive performance of the model. The experimental results for the mouse dataset exhibit similar trends. In Figure 3H, when the gene sequence length is 200bp, the ACC value for 3-mer is the highest, reaching 93.7%. Furthermore, in Figure 3F, the MCC and AUC values for 3-mer are also higher than for other k-mers. Consequently, based on the aforementioned results, the optimal performance for predicting DNA enhancers is achieved through the feature representation generated by 3-mer at a gene sequence length of 200bp. This also indicates the potential to optimize the model's ability in identifying enhancers by adjusting its learning of k-mer features, catering to diverse applications across different scenarios.

Impact of different feature encoding methods on model performance

In recent years, advanced DNA sequence feature encodings such as One-hot encoding, Nucleotide Chemical Property (NCP)-based encoding, and Dinucleotide Physicochemical Property (DPCP)-based encoding have become common in aiding machine learning models to process DNA sequences. These feature codes have found applications in bioinformatics research, including enhancer prediction. In this study, we generated these features and compared them with the 3-mer POCD-ND codes to identify the optimal sequence coding method for model performance. To ensure accurate results, each set of models was trained ten times using both human and mouse datasets with sequence lengths of 200 bp, and the experimental results are presented in Figure 4.

From Figures 4A and 4E, it can be observed that the average SN values for the One-hot, NCP, and DPCP feature encodings in both the human and mouse datasets all exceed 85%. However, their SP values are relatively low, and the average values from ten training experiments are inferior to those of the POCD-ND encoding. The ACC values of the POCD-ND encoding are generally distributed around 93%, consistently exceeding 90% overall. This indicates that the POCD-ND encoding can extract more distinctive features of DNA enhancers, effectively aiding the model in enhancer prediction. Comparing MCC and AUC values, POCD-ND encoding consistently outperforms other feature encodings in recognizing enhancers for both human and mouse datasets. POCD-ND feature encoding captures the distribution of nucleotides with different k-mers at specific positions in the sequence, and PSTNPss assigns appropriate scores to different levels of k-mers, making them more discriminative. In contrast, sequence encoders based on physicochemical properties often overlook the

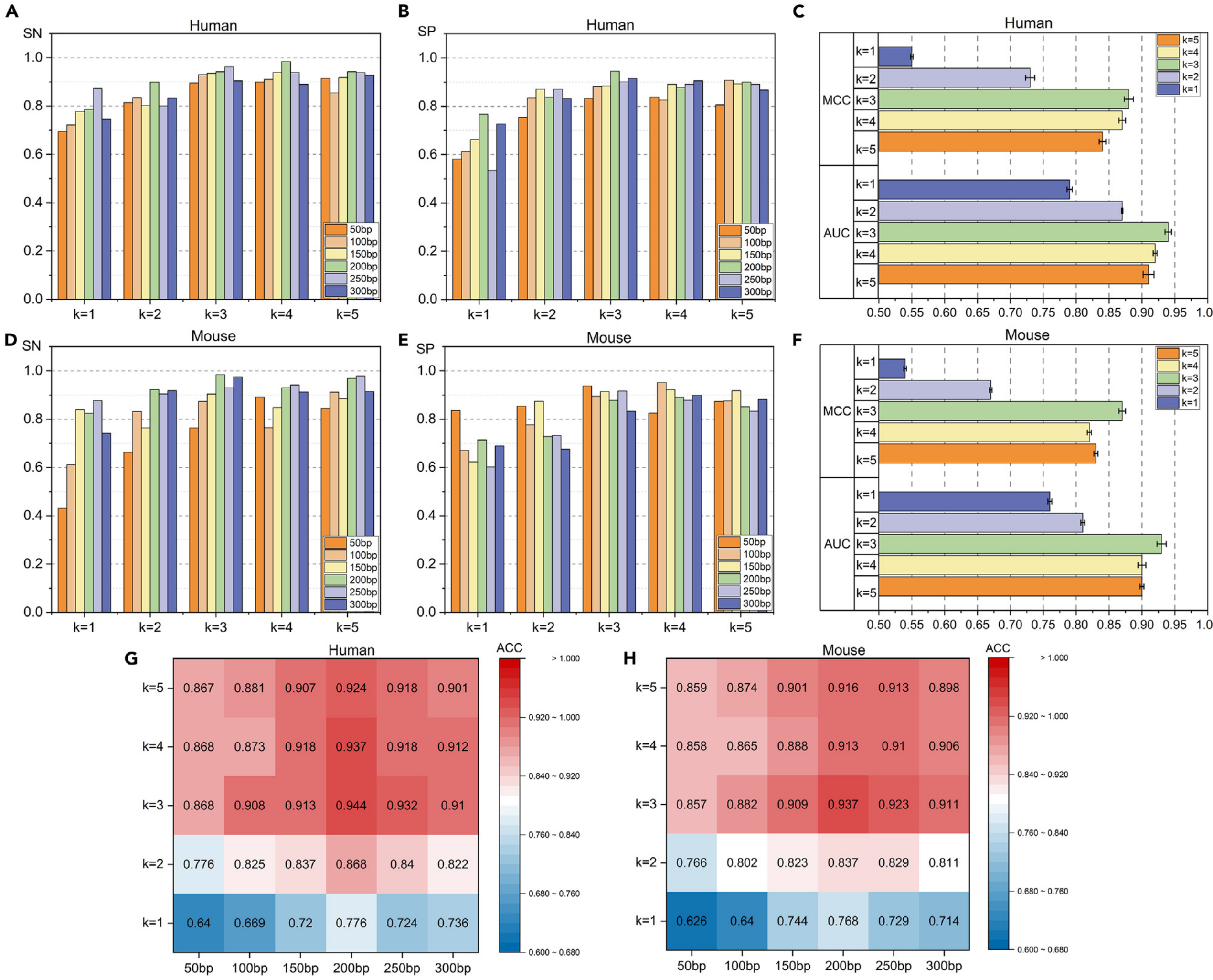


Figure 3. Experimental results on model performance with varied sequence lengths and k-mers

(A), (B), and (G): Evaluation results of SN, SP, and ACC values for experiments on the human dataset, respectively.(D), (E), and (H): Evaluation results of SN, SP, and ACC values for experiments on the mouse dataset, respectively.(C) and (F): Results of MCC value and AUC value assessment for different k-mer experiments at a 200 bp sequence length for human and mouse datasets, respectively.

positional information of k-mers, and One-hot encoding focuses solely on the distributional information of 1-mer. These results suggest that existing encoders struggle to effectively capture the position-aware discriminative distribution of k-mers, while the POCD-ND encoder excels at capturing the nucleotide synthesis discriminative pattern of DNA enhancer sequences, significantly aiding in modeling the prediction of DNA enhancer tasks.

Comparison with classical machine learning methods

Given the extensive application of machine learning methods in predicting DNA sequence elements, we are comparing the convolutional neural network (CNN) model with five classical machine learning methods: random forest (RF), logistic regression (LF), k-Nearest Neighbors (KNN), support vector machine (SVM), and extreme gradient boosting (XGBoost). We flatten the 1 × 197-dimensional feature matrix encoded with 3-mer POCD-ND into vector form, serving as input for RF, LF, KNN, SVM, and XGBoost. Similarly, we repetitively train each set of models ten times using human and mouse datasets with a sequence length of 200 bp. The resulting evaluation outcomes are illustrated in Figure 5.

We calculated the averages of the evaluation results obtained after training all models ten times and summarized them in Table 2.

Figure 5A displays the distribution of SN values for the test human dataset. It can be observed that LR and KNN show relatively scattered distributions, while SVM and XGBoost present more concentrated distributions. Particularly, XGBoost exhibits significantly higher SN values than the CNN model. However, as indicated in Figure 5B, the situation regarding SP values is less favorable. This suggests that for imbalanced datasets with unequal proportions of positive and negative samples, machine learning linear models struggle to effectively learn the

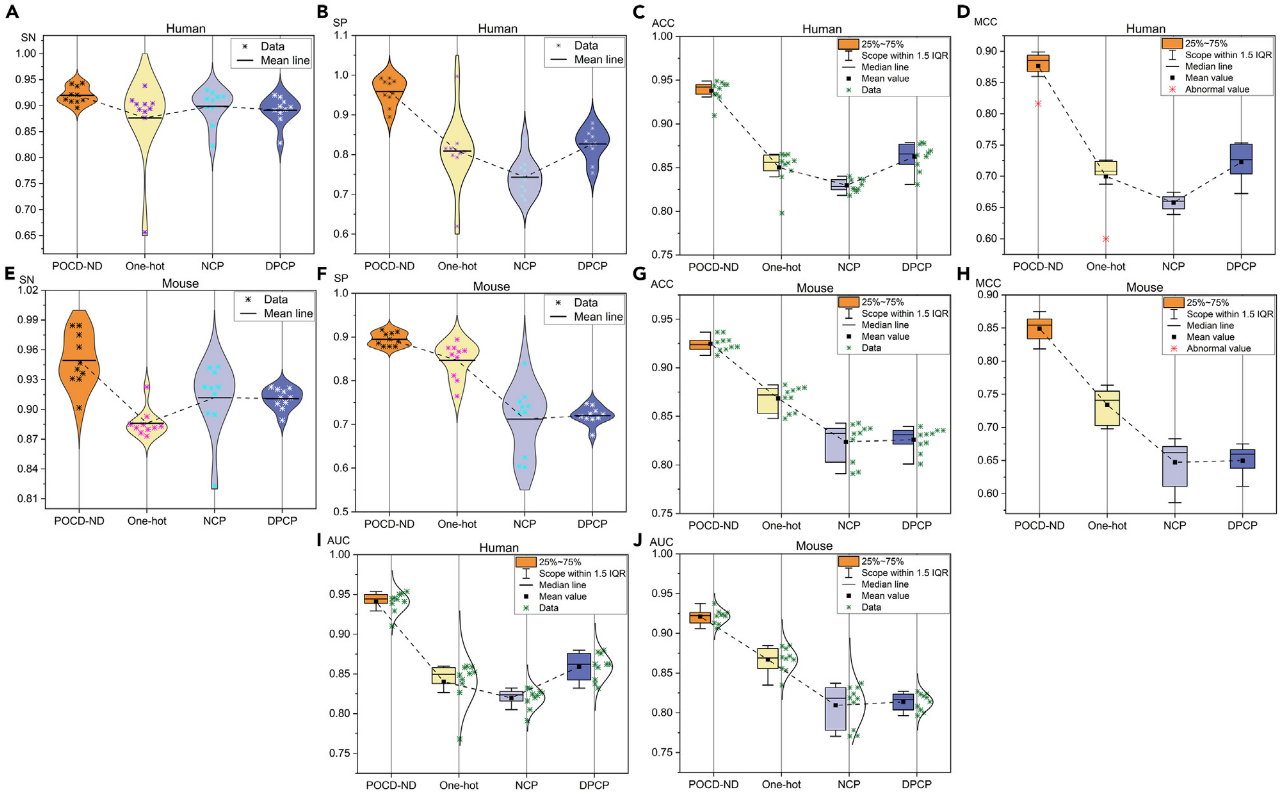


Figure 4. Experimental results on the impact of various sequence feature encodings on model performance over ten training sessions

(A), (B), (C), (D), and (I): Evaluation results of SN, SP, ACC, MCC, and AUC values for the human dataset.(E), (F), (G), (H), and (J): Evaluation results of SN, SP, ACC, MCC, and AUC values for the mouse dataset.

distinguishing features within them. The CNN models for testing both the human and mouse datasets have higher ACC values compared to ML models, averaging 93.8% and 92.5%, respectively. Employing the POCD-ND feature encoding for feature extraction, the relative advantage of the CNN model becomes more pronounced in scenarios of sparse positive data in actual datasets. Its capability in predicting DNA enhancers surpasses that of other machine learning models. When comparing MCC values and AUC values, it is observed that LR, RF, and SVM achieved similar performance on the mouse dataset, while the XGBoost model outperformed the other four ML models. In summary, these results suggest that the CNN model demonstrates better performance and is more effective in learning features extracted by the POCD-ND encoder, leading to more accurate predictions of DNA enhancers.

Comparison with existing DNA enhancer predictors

To demonstrate the generalizability and effectiveness of the proposed method in recognizing DNA enhancers, we are comparing the PDCNN model with established state-of-the-art DNA enhancer predictors. The encoder of the PDCNN model selects 3-mers for encoding. It is common for various models, when trained or tested on different datasets, to produce diverse recognition outcomes. To impartially evaluate prediction performance, we conducted a comparative analysis using the dataset created by Liu et al.[21] This experiment comprises two phases, as outlined in Figure 6.

In the first layer, the model determines whether the DNA gene sequence is an enhancer or non-enhancer. If it is recognized as the former, the model then proceeds to the second layer to evaluate the strength of the enhancer. The details of these comparative models are outlined in Table 3. It's important to highlight that the iEnhancer-ELM model is exclusively developed for the recognition of DNA enhancers, excluding an evaluation of their strength. Consequently, during the comparison, the second layer involving iEnhancer-ELM is disregarded.

The above methods all used Liu's dataset to construct the model, trained these methods along with PDCNN, and then evaluated them on an independent test dataset, with the results shown in Figure 7.

Despite a reduction in the models' classification effects in the second layer compared to the first layer, the superior performance of PDCNN over existing methods remains evident. Figure 7 illustrates that the SN and SP values of each model in the first layer are similar, indicating their effective differentiation between enhancers and non-enhancers. However, in the intensity analysis, iEnhancer-2L's SN value is only 40% or more, indicating poor performance in identifying strong enhancers. In contrast, PDCNN demonstrates superior performance in both identifying enhancers and determining their strengths, with ACC values of 96.3% and 94.2% in both stages.

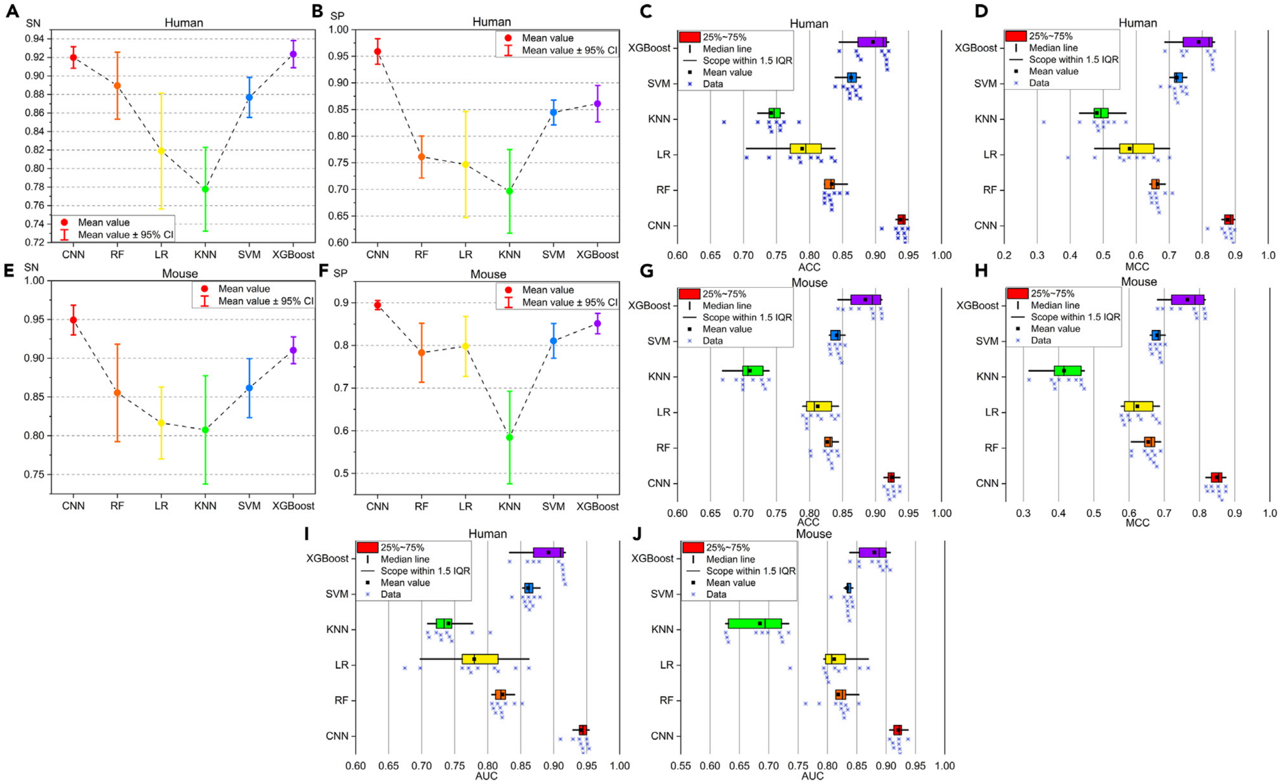


Figure 5. Experimental results from ten training sessions comparing ML and DL models

(A), (B), (C), (D), and (I): Evaluation results of SN, SP, ACC, MCC, and AUC values for the human dataset. (E), (F), (G), (H), and (J): Evaluation results of SN, SP, ACC, MCC, and AUC values for the mouse dataset.

MCC values, reflecting the classification performance, reached 92.6% and 88.5% for PDCNN, achieving the highest accuracy and MCC values. This suggests that the PDCNN model exhibits stable and effective performance in the task of recognizing and classifying DNA enhancers.

Model interpretation and visualization

To validate the model's generalization ability, we downloaded enhancer sequences for the species *Drosophila melanogaster* from EnhancerAtlas 2.0, including gene sequences from various cell types such as Kc167, BG3, and OSC. Sequences were partitioned into

Table 2. Average of the results of the assessment of each model

Dataset	Model	SN	SP	ACC	MCC	AUC
Human	CNN	0.92	0.959	0.938	0.877	0.94
	RF	0.89	0.761	0.834	0.665	0.822
	LR	0.819	0.747	0.789	0.58	0.78
	KNN	0.778	0.697	0.742	0.48	0.741
	SVM	0.877	0.845	0.863	0.723	0.861
	XGBoost	0.924	0.861	0.896	0.789	0.892
Mouse	CNN	0.949	0.895	0.925	0.849	0.921
	RF	0.855	0.783	0.827	0.655	0.818
	LR	0.817	0.798	0.812	0.624	0.811
	KNN	0.808	0.584	0.71	0.415	0.685
	SVM	0.862	0.811	0.841	0.68	0.833
	XGBoost	0.91	0.852	0.884	0.765	0.88

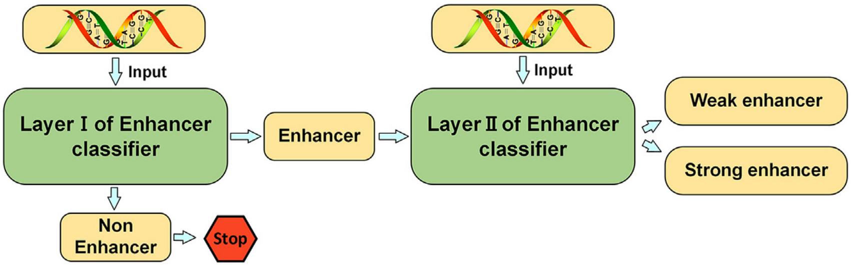


Figure 6. Comparison of experimental model workflow

200bp lengths based on the conclusions drawn from the above experiments. In the training dataset, the ratio of positive to negative samples for enhancers was 2865:9536, while in the testing dataset, the ratio was 478:1303. We compared and evaluated the performance using the PDCNN model against several enhancer recognition models, namely Sbper,[19] Biren,[20] PEDH[34] and iEnhancer-BERT[36] (model details can be found in the introduction section). The encoder for the PDCNN model employed a 3-mer encoding. To better interpret the performance of the PDCNN model, we extracted and visualized the input and output of all network layers, including the original input, the output of the first convolutional layer, the output of the second convolutional layer, and the output of the fully connected layer. To facilitate understanding these features, UMAP was used to visualize the distribution of positive and negative samples. The results and visualizations are presented in Figure 8.

As shown in Figure 8A, PDCNN demonstrates excellent performance in recognizing *D. melanogaster*, achieving an ACC value of 91.2%, the highest recognition accuracy among the various models. The AUC value is only slightly lower than that of PEDH, indicating the model's strong generalization ability in recognizing new datasets. In terms of model interpretation, the features of negative and positive samples in the raw input are indistinguishable in Figure 8B. However, as PDCNN learns layer by layer, based on the output features of the first and second convolutional layers, a progressively clear separation of positive and negative sample boundaries can be observed in the feature space. The output of the fully connected layer ultimately distinctly separates each sample. These results demonstrate PDCNN's ability to learn from the POCD-ND encoder and extract latent features, enabling more decisive recognition of DNA enhancers.

Conclusions

To enhance the efficiency and accuracy of DNA enhancer identification, we introduce a deep learning-based prediction model called PDCNN. Comparative experiments reveal that PDCNN surpasses state-of-the-art machine learning methods and existing models in the identification of DNA enhancers. The model's superior performance is attributed to the encoder, which provides more discriminative features, allowing the deep learning model to leverage its powerful learning capabilities effectively. The POCD-ND encoder captures the class density difference of k-mer incidence at specific locations, statistically regularizing it by its minimum value and assigning effective scores to k-mers. This approach better expresses informative features of DNA enhancer sequences, feeding them into the convolutional neural network for learning. However, in the human genome, enhancer sequences are comparatively rare and exhibit substantial positional

Table 3. Existing advanced DNA enhancer predictors

Model	Description	Source
iEnhancer-2L	Sequences are feature encoded using the PseKNC method and fed into a support vector machine SVM for recognition	Liu et al.[21]
iEnhancer-ECNN	Sequences are processed using One-hot coding and k-mers, and integrated models are constructed using CNNs	Nguyen et al.[24]
iEnhancer-XG	Combining the five coding features of the sequence and using XGBoost as the base classifier	Cai et al.[25]
iEnhancer-EBLSTM	Input DNA sequences were encoded using 3-mer and then enhancers were predicted by bidirectional LSTM	Niu et al.[26]
iEnhancer-DCSV	Combining One-hot coding and NCP coding for prediction using improved DenseNet and improved CBAM attention module	Jia et al.[27]
iEnhancer-DCSA	Combining n-gram segmentation with skip-gram segmentation, employing a dual-scale fusion convolutional neural network.	Wang et al.[28]
iEnhancer-GAN	Building a CNN architecture by combining word embedding skip-gram and sequence generation adversarial networks	Bao et al.[31]
Enhancer-LSTMAtt	After simple encoding of sequences, using Bi-LSTM and attention-based deep learning methods	Huang et al.[33]
iEnhancer-ELM	A enhancer recognition method based on a BERT-like enhancer language model, DNABERT.	Li et al.[35]

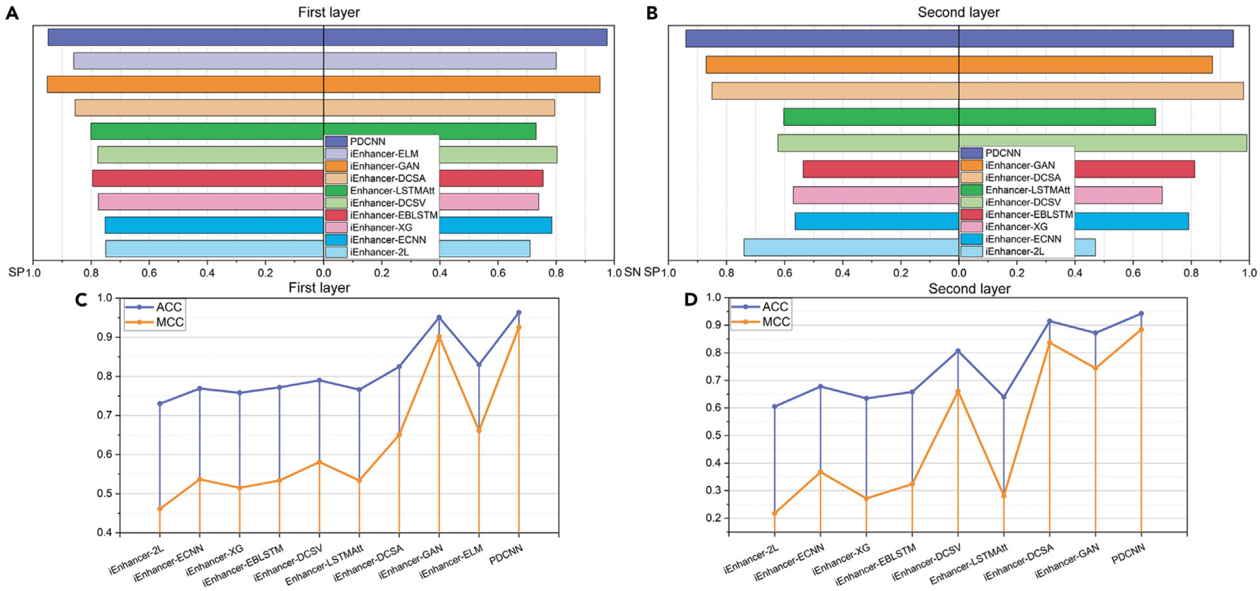


Figure 7. Experimental results for each comparison model

(A) SN and SP values of the model for identifying enhancers at the first layer.
(B) SN and SP values of the model at the second layer of intensity analysis.
(C) Results of the model's ACC and MCC evaluations at the first layer.
(D) Results of the model's ACC and MCC evaluations at the second layer.

variability. Future research should address these challenges more effectively and propose corresponding enhancements to improve the model's accuracy and applicability. Furthermore, it has been shown that highly methylated DNA regions affect the regulation of genes by nearby enhancers.[43] Therefore future research may take DNA enhancer recognition in combination with epigenetic modification of methylation as an important direction.

Limitations of the study

While the PDCNN model exhibits strengths, there are still certain limitations. While the PDCNN model exhibits strengths, there are still certain limitations. The optimal k-mer length for the model's encoder is determined through experiments on specific datasets. Although it performs well on other datasets, the reliance on manually defined rules introduces uncertainty. An avenue for future improvement involves enabling the model to autonomously learn the optimal k-mer length during the recognition process, thereby enhancing result accuracy. Additionally, it is worth noting that the dataset utilized in this study is balanced across classes.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- METHOD DETAILS
  - Datasets
  - The position-aware encoding of positive and negative modification classes of nucleotides
  - A model of convolutional neural network based on position awareness of positively and negatively modified nucleotide classes
- QUANTIFICATION AND STATISTICAL ANALYSIS
  - Model performance evaluation metrics

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2024.110030.

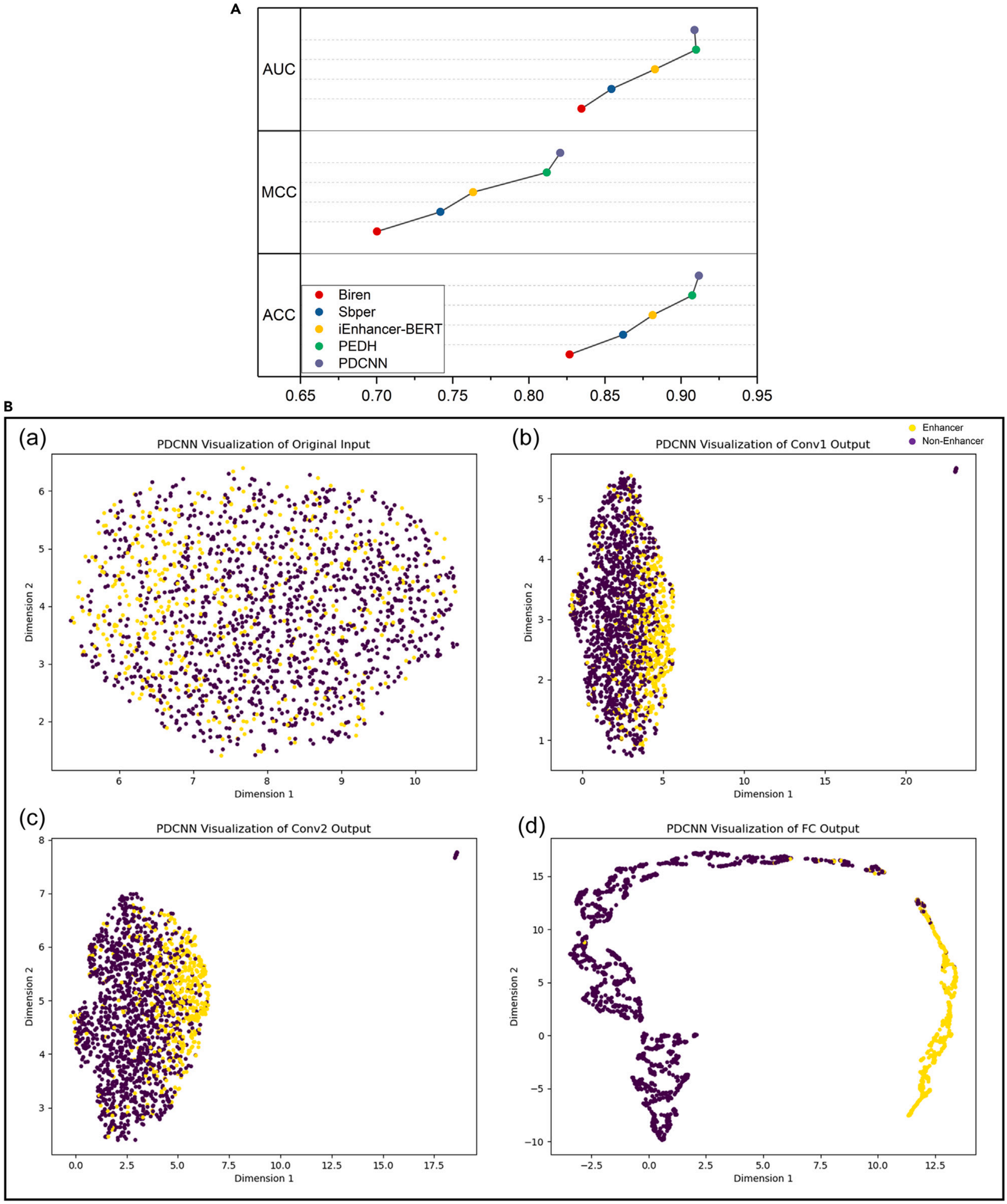


Figure 8. The visualization results of the model recognizing the *D. melanogaster* dataset

(A) Comparative results of the model recognizing the *D. melanogaster* dataset.

(B) Distribution of positive and negative samples in the 2D feature space.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support from the National Natural Science Foundation of China (Grant Numbers: 51663001, 52063002, 42061067 and 61741202).

AUTHOR CONTRIBUTIONS

Conceptualization: H.W., L.M., and L.Y.
Methodology: H.W.
Investigation: H.W., L.Y., W.Y., L.M., and G.L.
Formal analysis: H.W., L.Y., and W.Y.
Resources: L.M. and G.L.
Data curation: H.W. and L.Y.
Writing – original draft: H.W.
Writing – review and editing: H.W., L.M., W.Y., L.Y., and G.L.
Visualization: H.W.
Supervision: L.M. and G.L.
Funding acquisition: L.M. and G.L.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: April 8, 2024
Revised: April 23, 2024
Accepted: May 16, 2024
Published: May 19, 2024

REFERENCES

1. Tsujimura, T., Takase, O., Yoshikawa, M., Sano, E., Hayashi, M., Hoshi, K., Takato, T., Toyoda, A., Okano, H., and Hishikawa, K. (2020). Controlling gene activation by enhancers through a drug-inducible topological insulator. Elife *9*, e47980.
2. Peng, Y., and Zhang, Y. (2018). Enhancer and super-enhancer: Positive regulators in gene transcription. Animal Model. Exp. Med. *1*, 169–179.
3. Li, K., Liu, Y., Cao, H., Zhang, Y., Gu, Z., Liu, X., Yu, A., Kaphle, P., Dickerson, K.E., Ni, M., and Xu, J. (2020). Interrogation of enhancer function by enhancer-targeting CRISPR epigenetic editing. Nat. Commun. *11*, 485.
4. Jia, Y., Chng, W.-J., and Zhou, J. (2019). Super-enhancers: critical roles and therapeutic targets in hematologic malignancies. J. Hematol. Oncol. *12*, 77.
5. Carelli, F.N., Liechti, A., Halbert, J., Warnefors, M., and Kaessmann, H. (2018). Repurposing of promoters and enhancers during mammalian evolution. Nat. Commun. *9*, 4066.
6. Zhang, G., Shi, J., Zhu, S., Lan, Y., Xu, L., Yuan, H., Liao, G., Liu, X., Zhang, Y., Xiao, Y., and Li, X. (2018). DiseaseEnhancer: a resource of human disease-associated enhancer catalog. Nucleic Acids Res. *46*, D78–D84.
7. Corradin, O., and Scacheri, P.C. (2014). Enhancer variants: evaluating functions in common disease. Genome Med. *6*, 85.
8. Boyd, M., Thodberg, M., Vitezic, M., Bornholdt, J., Vitting-Seerup, K., Chen, Y., Coskun, M., Li, Y., Lo, B.Z.S., Klausen, P., et al. (2018). Characterization of the enhancer and promoter landscape of inflammatory bowel disease from human colon biopsies. Nat. Commun. *9*, 1661.
9. Lyu, Y., Zhang, Z., Li, J., He, W., Ding, Y., and Guo, F. (2021). iEnhancer-KL: A novel two-layer predictor for identifying enhancers by position specific of nucleotide composition. IEEE ACM Trans. Comput. Biol. Bioinf *18*, 2809–2815.
10. Woolfe, A., Goodson, M., Goode, D.K., Snell, P., McEwen, G.K., Vavouri, T., Smith, S.F., North, P., Callaway, H., Kelly, K., et al. (2005). Highly conserved non-coding sequences are associated with vertebrate development. PLoS Biol. *3*, e7.
11. Pennacchio, L.A., Ahituv, N., Moses, A.M., Prabhakar, S., Nobrega, M.A., Shoukry, M., Minovitsky, S., Dubchak, I., Holt, A., Lewis, K.D., et al. (2006). In vivo enhancer analysis of human conserved non-coding sequences. Nature *444*, 499–502.
12. Mora, A., Sandve, G.K., Gabrielsen, O.S., and Eskeland, R. (2016). In the loop: promoter–enhancer interactions and bioinformatics. Briefings Bioinf. *17*, 980–995.
13. Moore, J.E., Pratt, H.E., Purcaro, M.J., and Weng, Z. (2020). A curated benchmark of enhancer-gene interactions for evaluating enhancer-target gene prediction methods. Genome Biol. *21*, 17.
14. Liu, B., Li, K., Huang, D.-S., and Chou, K.-C. (2018). iEnhancer-EL: identifying enhancers and their strength with ensemble learning approach. Bioinformatics *34*, 3835–3842.
15. Fernandez, M., and Miranda-Saavedra, D. (2012). Genome-wide enhancer prediction from epigenetic signatures using genetic algorithm-optimized support vector machines. Nucleic Acids Res. *40*, e77.
16. Firpi, H.A., Ucar, D., and Tan, K. (2010). Discover regulatory DNA elements using chromatin signatures and artificial neural network. Bioinformatics *26*, 1579–1586.
17. Rajagopal, N., Xie, W., Li, Y., Wagner, U., Wang, W., Stamatoyannopoulos, J., Ernst, J., Kellis, M., and Ren, B. (2013). RFECS: a random-forest based algorithm for enhancer identification from chromatin state. PLoS Comput. Biol. *9*, e1002968.
18. Erwin, G.D., Oksenberg, N., Truty, R.M., Kostka, D., Murphy, K.K., Ahituv, N., Pollard, K.S., and Capra, J.A. (2014). Integrating diverse datasets improves developmental enhancer prediction. PLoS Comput. Biol. *10*, e1003677.
19. Singh, A.P., Mishra, S., and Jabin, S. (2018). Sequence based prediction of enhancer regions from DNA random walk. Sci. Rep. *8*, 15912.
20. Yang, B., Liu, F., Ren, C., Ouyang, Z., Xie, Z., Bo, X., and Shu, W. (2017). BiRen: predicting enhancers with a deep-learning-based model using the DNA sequence alone. Bioinformatics *33*, 1930–1936. https://doi.org/10.1093/bioinformatics/btx105.
21. Liu, B., Fang, L., Long, R., Lan, X., and Chou, K.C. (2016). iEnhancer-2L: a two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition. Bioinformatics *32*, 362–369. https://doi.org/10.1093/bioinformatics/btv604.
22. Liu, B. (2016). iEnhancer-PsedeKNC: Identification of enhancers and their subgroups based on Pseudo degenerate kmer nucleotide composition. Neurocomputing *217*, 46–52.
23. Jia, C., and He, W. (2016). EnhancerPred: a predictor for discovering enhancers based on the combination and selection of multiple features. Sci. Rep. *6*, 38741.
24. Nguyen, Q.H., Nguyen-Vo, T.-H., Le, N.Q.K., Do, T.T.T., Rahardja, S., and Nguyen, B.P.

(2019). iEnhancer-ECNN: identifying enhancers and their strength using ensembles of convolutional neural networks. BMC Genom. *20*, 951.

25. Cai, L., Ren, X., Fu, X., Peng, L., Gao, M., and Zeng, X. (2021). iEnhancer-XG: interpretable sequence-based enhancers and their strength predictor. Bioinformatics *37*, 1060–1067.
26. Niu, K., Luo, X., Zhang, S., Teng, Z., Zhang, T., and Zhao, Y. (2021). iEnhancer-EBLSTM: Identifying Enhancers and Strengths by Ensembles of Bidirectional Long Short-Term Memory. Front. Genet. *12*, 665498. https://doi.org/10.3389/fgene.2021.665498.
27. Jia, J., Lei, R., Qin, L., Wu, G., and Wei, X. (2023). iEnhancer-DCSV: Predicting enhancers and their strength based on DenseNet and improved convolutional block attention module. Front. Genet. *14*, 1132018. https://doi.org/10.3389/fgene.2023.1132018.
28. Wang, W., Wu, Q., and Li, C. (2023). iEnhancer-DCSA: identifying enhancers via dual-scale convolution and spatial attention. BMC Genom. *24*, 393. https://doi.org/10.1186/s12864-023-09468-1.
29. Basith, S., Hasan, M.M., Lee, G., Wei, L., and Manavalan, B. (2021). Integrative machine learning framework for the identification of cell-specific enhancers from the human genome. Briefings Bioinf. *22*, bbab252. https://doi.org/10.1093/bib/bbab252.
30. Le, N.Q.K., Ho, Q.-T., Nguyen, T.-T.-D., and Ou, Y.-Y. (2021). A transformer architecture based on BERT and 2D convolutional neural network to identify DNA enhancers from sequence information. Briefings Bioinf. *22*, bbab005. https://doi.org/10.1093/bib/bbab005.
31. Bao, W., Yang, B., and Chen, B. (2021). 2-hydr_ensemble: lysine 2-hydroxyisobutyrylation identification with ensemble method. Chemometr. Intell. Lab. Syst. *215*, 104351.
32. Alakuş, T.B. (2023). A Novel Repetition Frequency-Based DNA Encoding Scheme to Predict Human and Mouse DNA Enhancers with Deep Learning. Biomimetics *8*, 218. https://doi.org/10.3390/biomimetics8020218.
33. Huang, G., Luo, W., Zhang, G., Zheng, P., Yao, Y., Lyu, J., Liu, Y., and Wei, D.Q. (2022). Enhancer-LSTMAtt: A Bi-LSTM and Attention-Based Deep Learning Method for Enhancer Recognition. Biomolecules *12*, 995. https://doi.org/10.3390/biom12070995.
34. Kaur, A., Chauhan, A.P.S., and Aggarwal, A.K. (2023). Prediction of enhancers in dna sequence data using a hybrid cnn-dlstm model. IEEE ACM Trans. Comput. Biol. Bioinf *20*, 1327–1336.
35. Li, J., Wu, Z., Lin, W., Luo, J., Zhang, J., Chen, Q., and Chen, J. (2023). iEnhancer-ELM: improve enhancer identification by extracting position-related multiscale contextual information based on enhancer language models. Bioinform. Adv. *3*, vbad043. https://doi.org/10.1093/bioadv/vbad043.
36. Luo, H., Chen, C., Shan, W., Ding, P., and Luo, L. (2022). iEnhancer-BERT: A Novel Transfer Learning Architecture Based on DNA-Language Model for Identifying Enhancers and Their Strength (Springer), pp. 153–165.
37. Smith, G.D., Ching, W.H., Cornejo-Páramo, P., and Wong, E.S. (2023). Decoding enhancer complexity with machine learning and high-throughput discovery. Genome Biol. *24*, 116. https://doi.org/10.1186/s13059-023-02955-4.
38. Liang, Y., Zhang, S., Qiao, H., and Cheng, Y. (2021). iEnhancer-MFGBDT: Identifying enhancers and their strength by fusing multiple features and gradient boosting decision tree. Math. Biosci. Eng. *18*, 8797–8814. https://doi.org/10.3934/mbe.2021434.
39. Chen, S., Gan, M., Lv, H., and Jiang, R. (2021). DeepCAPE: A Deep Convolutional Neural Network for the Accurate Prediction of Enhancers. Dev. Reprod. Biol. *19*, 565–577. https://doi.org/10.1016/j.gpb.2019.04.006.
40. Mills, C., Marconett, C.N., Lewinger, J.P., and Mi, H. (2023). PEACOCK: a machine learning approach to assess the validity of cell type-specific enhancer-gene regulatory relationships. NPJ Syst. Biol. Appl. *9*, 9. https://doi.org/10.1038/s41540-023-00270-z.
41. Chen, Z., Zhang, J., Liu, J., Dai, Y., Lee, D., Min, M.R., Xu, M., and Gerstein, M. (2021). DECODE: a Deep-learning framework for Condensing enhancers and refining boundaries with large-scale functional assays. Bioinformatics *37*, i280–i288. https://doi.org/10.1093/bioinformatics/btab283.
42. MacPhillamy, C., Alinejad-Rokny, H., Pitchford, W.S., and Low, W.Y. (2022). Cross-species enhancer prediction using machine learning. Genomics *114*, 110454. https://doi.org/10.1016/j.ygeno.2022.110454.
43. Angeloni, A., and Bogdanovic, O. (2019). Enhancer DNA methylation: implications for gene regulation. Essays Biochem. *63*, 707–715.
44. Visel, A., Minovitsky, S., Dubchak, I., and Pennacchio, L.A. (2007). VISTA Enhancer Browser—a database of tissue-specific human enhancers. Nucleic Acids Res. *35*, D88–D92.
45. Fu, L., Niu, B., Zhu, Z., Wu, S., and Li, W. (2012). CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics *28*, 3150–3152.
46. Zhang, T., Tang, Q., Nie, F., Zhao, Q., and Chen, W. (2022). DeepLncPro: an interpretable convolutional neural network model for identifying long non-coding RNA promoters. Briefings Bioinf. *23*, bbac447. https://doi.org/10.1093/bib/bbac447.
47. Nabeel Asim, M., Ali Ibrahim, M., Fazeel, A., Dengel, A., and Ahmed, S. (2023). DNA-MP: a generalized DNA modifications predictor for multiple species based on powerful sequence encoding method. Briefings Bioinf. *24*, bbac546.
48. Liao, M., Zhao, J.P., Tian, J., and Zheng, C.H. (2022). iEnhancer-DCLA: using the original sequence to identify enhancers and their strength based on a deep learning framework. BMC Bioinf. *23*, 480. https://doi.org/10.1186/s12859-022-05033-x.
49. Ho, Y., and Wookey, S. (2020). The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling. IEEE Access *8*, 4806–4813.
50. Lobo, J.M., Jiménez-Valverde, A., and Real, R. (2008). AUC: a misleading measure of the performance of predictive distribution models. Global Ecol. Biogeogr. *17*, 145–151.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Human and Mouse Dataset	VISTA Enhancer Browser	https://enhancer.lbl.gov/
D.melanogaster Dataset	EnhancerAtlas 2.0	http://www.enhanceratlas.org/
Software and algorithms		
PDCNN	This paper	https://github.com/xing1999/PDCNN
Python v3.8	Python	https://www.python.org/
NumPy v1.17.0	NumPy	https://numpy.org/
PyTorch v1.2.0	PyTorch	https://pytorch.org/
scikit-learn v0.0	scikit-learn	https://scikit-learn.org/stable/

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact – Li Mengshan, msli@gnnu.edu.cn.

Materials availability

This study did not generate new unique reagents.

Data and code availability

- Data: All the enhancer data used in this study are available free of charge at GitHub. (https://github.com/xing1999/PDCNN).
- Code: Our code is publicly accessible at https://github.com/xing1999/PDCNN.
- Other: Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

METHOD DETAILS

Datasets

We obtained the latest positive and negative sequence data of DNA enhancers for both human (hg19/GRCh37) and mouse (mm9/NCBIM37) from the VISTA Enhancer Browser,[44] an online tool designed to assist researchers in exploring and analyzing enhancer elements in vertebrate genomes. This platform provides a database of enhancer recognition candidate sequences. To avoid redundancy and reduce homology bias while better preserving the original distribution, we employed the CD-HIT program[45] to eliminate sequences with over 30% sequence similarity. For experimental purposes, we categorized the dataset into two classes, human (hg19) and mouse (mm9), further dividing it into training, validation, and test sets in an 8:1:1 ratio. Throughout the training process, the validation set, derived from the training set, was employed to fine-tune the model's hyperparameters and detect overfitting. The test set was reserved for the final evaluation of the model's performance post-training.The breakdown of the benchmark dataset is as follows:

$$\mathbb{S} = \mathbb{S}^{+} \cup \mathbb{S}^{-} \tag{Equation 1}$$

where $\mathbb{S}$ represents the complete dataset of a species, $\mathbb{S}^{+}$ signifies the positive data subset containing enhancers, and $\mathbb{S}^{-}$ represents the negative data subset. The union of these two subsets is denoted as $\cup$.

To ensure a fair comparison with prior studies, we incorporated the dataset established by Liu et al.,[21] a widely utilized resource in enhancer prediction research. Liu et al. compiled enhancer sequences from nine distinct cell lines, extracting equal-length 200 bp fragments. They applied the CD-HIT software[45] to eliminate sequences with a similarity exceeding 80%. The dataset was subsequently divided into two segments: the training dataset for model training and the independent test dataset for model testing.The training dataset comprises 1484 samples, consisting of 742 strong enhancers, 742 weak enhancers, and 1484 non-enhancer samples. The test dataset includes 200 samples, featuring 100 strong enhancers, 100 weak enhancers, and 200 non-enhancer samples. Specific information about the utilized dataset is detailed in Table S1.

The position-aware encoding of positive and negative modification classes of nucleotides

Deep learning models typically cannot process raw DNA sequences directly due to their reliance on numerical inputs. DNA sequences, composed of four basic nucleotides, need to be transformed into numerical values for use in models. Various encoding methods have been proposed for this purpose, with common ones including One-hot encoding,[27,46] NCP encoding,[27,46] and DPCP encoding.[29,46] However, these encoders still fall short in fully capturing the positional feature information of nucleotides in DNA sequences.To generate a more comprehensive numerical representation of DNA sequences and capture the distribution patterns of nucleotides at different positions, we have adopted a position-aware K-mer coding method based on the normalized difference of positive and negative modification class densities (POCD-ND).[47]

To convert a DNA sequence into a numerical representation, k-mers are generated by sliding a window of a fixed size k with a specific step size. In this process, the DNA sequence is divided into sub-sequences, each referred to as a k-mer, representing a set of nucleotides. The size of the k-mer, or sub-sequence, depends on the size of the window in which they were generated. All independent k-mers in a subsequence are collected to form a vocabulary. The size of the lexicon depends on the size of \(k\), assuming there are enough samples of sequences.For example, when $k = 1$, the size of the vocabulary base is $4^1 = 4$, corresponding to A, C, G and T. When $k = 2$, the size of the vocabulary base is $4^2 = 16$, containing 16 types of k-mers such as AA, AC, AG, and AT, etc. In each sequence, the position of generating k-mers can be denoted by $P_i = P_1, P_2 \cdots$. As the size of the k-mers increases, the size of the lexicon increases, and as the length of the sequence increases, the number of positions increases accordingly.

After splitting the sequences into k-mers, the frequencies of the lexicon at different positions in each sequence were counted separately for the positive and negative sequences. These counts were then organized into two matrices, A^{pos} and A^{neg}, where the dimensions of the matrices are z positions and n words, as shown in Equation 2.

$$A^{pos} = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{z,1} & f_{z,2} & \cdots & f_{z,n} \end{bmatrix}, A^{neg} = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{z,1} & f_{z,2} & \cdots & f_{z,n} \end{bmatrix} \quad \text{(Equation 2)}$$

where each item f_(i,j) of the matrix is represented as the frequency of the jth word at the ith position of the sequence.

After obtaining the frequency distribution matrices for the positive and negative class sequences, the total number of k-mers in the positive and negative sequences, denoted as NS, is further counted. The frequency distribution matrices are then normalized to obtain their density distribution matrices, A^{posden} and A^{negden}, as shown in Equation 3.

$$\begin{cases} A^{posden} = \dfrac{A^{pos}}{NS^{pos}}, 0 \le A^{posden} \le 1 \\ A^{negden} = \dfrac{A^{neg}}{NS^{neg}}, 0 \le A^{posneg} \le 1 \end{cases} \quad \text{(Equation 3)}$$

To achieve a more comprehensive statistical representation of the DNA sequence, we proceeded to calculate the PSTNPss score for the jth lexical k-mer at the ith position in the gene sequence using the density distribution matrices of the positive and negative classes. Equation 4 provides a generalized mathematical expression to compute the PSTNPss score.

$$PSTNPss = A^{posden} - A^{negden} \quad \text{(Equation 4)}$$

The k-mer density distributions of positive and negative category sequences at the same position often differ, and may even be extremely different. In order to enhance the discriminative nature of the PSTNPss scores, category tradeoff values are introduced, resulting in the POCD-ND score matrix, as shown in Equation 5.

$$POCD - ND = \frac{PSTNPss}{\min\left(A^{posden}, A^{negden}\right)}, \min\left(A^{posden}, A^{negden}\right) > 0 \quad \text{(Equation 5)}$$

Significantly, when calculating POCD-ND fractions, the denominator becomes zero if one of the positive and negative class densities of a particular k-mer is zero, leading to the fraction becoming unsigned infinity. This situation rarely occurs,but to handle it effectively, a small nonzero value is introduced for division. In our experiments, this nonzero value was analyzed for performance with a step size of 0.1, ranging from 0.1 to 0.9. The experimental results revealed optimal performance when the minimum value was set to 0.1. Therefore, when the denominator is zero, the calculation is performed using 0.1.

A gene sequence of length L is divided into $L - k + 1$ k-mers. Subsequently, these k-mers are encoded into a $1 \times (L - k + 1)$ matrix corresponding to the POCD-ND score matrix in the order of their positions in the sequence. This encoded matrix is then fed into the model for learning. The gene sequence encoding process is illustrated in Figure S1.

A model of convolutional neural network based on position awareness of positively and negatively modified nucleotide classes

In contemporary times, most DNA gene sequence data are processed using recurrent neural network architectures such as LSTM, GRU, etc.[26,33,48] However, the output of the POCD-ND encoder is a feature matrix containing information about the distribution of nucleotides at specific positions. In reality, DNA enhancer sequences are only related to the information within their neighboring windows. To more effectively extract the hidden information in the feature matrix, this paper employs a convolutional neural network to process the encoded input feature matrix for the task of predicting DNA enhancers. A convolutional neural network model based on the POCD-ND encoder, referred to as PDCNN, is constructed using the PyTorch framework in the Python package. The specific network architecture is illustrated in Figure 2, incorporating a series of convolutional layers, a maximal pooling layer for extracting position-aware features independent of encoding spatial transformations, and a fully-connected layer for nonlinearly processing the extracted information from the upstream convolutional layers.

The PDCNN distinguishes itself from previous CNNs by having an input that includes location-specific distribution-aware information of k-mers in enhancer and non-enhancer sequences. The convolution operation in PDCNN is akin to extracting motifs from sequences with high activation feature information using a sliding window. Therefore, the PDCNN incorporates two 1D convolutional layers. The first convolutional layer is responsible for detecting motifs in the DNA enhancer sequences, while the second layer describes the associations between the motifs extracted on a larger scale. Following these convolutional layers, the model includes a fully connected layer, which integrates information from the entire sequence. Finally, the probabilities obtained using Sigmoid-type functions are utilized for prediction. The mathematical representation of the first convolutional layer is provided in Equation 6.

$$Conv(Z)_{i,j} = ReLU\left(\sum_{s=0}^{S-1}\sum_{n=0}^{N-1} W^j_{s,n} Z_{i+s,n}\right) \quad \text{(Equation 6)}$$

where Z represents the encoded feature matrix of the gene sequence, i is the index of the output position, and j is the index of the filter. Each convolutional filter W^j is an S×N matrix, where S is the filter size determined by hyperparameter optimization and N is the number of input channels. In the case of the first convolutional layer, N is the input dimension of the feature matrix after encoding the augmented subsequence. The ReLU function is expressed as:

$$ReLU = \begin{cases} x, \text{if } x \geq 0 \\ 0, \text{if } x < 0 \end{cases} \quad \text{(Equation 7)}$$

The objective of the position-aware k-mer feature coding fusion convolutional neural network is to establish a mapping relation:

$$\widehat{Y} = \arg\ \max f(\ POCD - ND_n(i); W) \quad \text{(Equation 8)}$$

where $\widehat{Y}$ represents the predicted result by the convolutional neural network for DNA enhancers; $POCD - ND_n$ denotes the feature matrix of DNA gene sequences encoded by POCD-ND; W is the parameter of the convolutional neural network, and f is the mapping function searched by the neural network.

To establish this mapping relationship, it is necessary to define a loss function that measures the difference between the predicted labels and the true labels. This loss function is then iteratively updated using gradient descent to minimize the overall loss. In this paper, a common cross-entropy loss function is employed,[49] formulated as follows:

$$L = -\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)} \log p^{(n)} + \left(1 - y^{(n)}\right) \log\left(1 - p^{(n)}\right)\right) \quad \text{(Equation 9)}$$

where N represents the sample capacity, $y^{(n)}$ is a binary variable, and $p^{(n)}$ is the prediction probability of the neural network for the n-th DNA-enhanced subsample.

QUANTIFICATION AND STATISTICAL ANALYSIS

Model performance evaluation metrics

To assess the classification performance of the model, commonly used metrics for classification performance are employed, consistent with previous evaluations. These metrics encompass sensitivity (SN), specificity (SP), accuracy (ACC), and Mathew's correlation coefficient (MCC). Additionally, the area under the working characteristic curve (AUC)[50] is utilized for evaluation. The specific calculation process for these evaluation metrics is outlined below.

$$\begin{cases} SN = \dfrac{TP}{TP+FN} \\ SP = \dfrac{TN}{TN+FP} \\ ACC = \dfrac{TP+TN}{TP+FN+TN+FP} \\ MCC = \dfrac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)\times(TP+FN)\times(TN+FP)\times(TN+FN)}} \\ AUC = \dfrac{\sum\limits_{i\in pos} rank_i - \dfrac{num_{pos}\left(num_{pos}+1\right)}{2}}{num_{pos}num_{neg}} \end{cases} \tag{Equation 10}$$

Where TP, TN, FP, and FN represent the number of samples with true positive, true negative, false positive, and false negative predictions, respectively. AUC (Area Under Curve) is defined as the area under the ROC curve enclosed by the axis. A value closer to 1.0 for AUC indicates better performance of the model.