DENIES: A deep learning based two-layer predictor for enhancing the identification of enhancers and their strength with DNA shape information

Ye Li[1], Chunquan Li[2], Jiquan Ma[1*]

[1] Department of Computer Science and Technology, Heilongjiang University, Harbin, China
[2] School of Medical informatics, Daqing Campus, Harbin Medical University, Daqing, China

* Corresponding author
Email address: majiquan@hlju.edu.cn

Abstract

The identification of enhancers has always been an important task in bioinformatics owing to their major role in regulating gene expression. For this reason, many computational algorithms devoted to enhancer identification have been put forward over the years, ranging from statistics and machine learning to the increasing popular deep learning. To boost the performance of their methods, more features tend to be extracted from the single DNA sequences and integrated to develop an ensemble classifier. Nevertheless, the sequence-derived features used in previous studies can hardly provide the 3D structure information of DNA sequences, which is regarded as an important factor affecting the binding preferences of transcription factors to regulatory elements like enhancers. Given that, we here propose DENIES, a deep learning based two-layer predictor for enhancing the identification of enhancers and their strength. Besides two common sequence-derived features (i.e. one-hot and *k*-mer), it introduces DNA shape for describing the 3D structures of DNA sequences. The results of performance comparison with a series of state-of-the-art methods conducted on the same datasets prove the effectiveness and robustness of our method. The code implementation of our predictor is freely available at https://github.com/hlju-liye/DENIES.

Keywords: Gene expression, Enhancer identification, 3D structure, Deep learning, Two-layer predictor, Strength, DNA shape

1. Introduction

The living and development of organisms are inseparable from the proper function of gene expression in cells, which is regulated by the concerted cooperation of various types of gene regulatory elements located in the non-coding regions of genome [1]. The typical regulatory elements include enhancers, promoters, silencers and insulators. Among these, enhancers are deemed as the most crucial ones responsible for regulating the transcription of their target genes. Different from gene proximal regulatory elements like promoters, the location of enhancers relative to their target genes can't be simply formulated. They can be located upstream or downstream and sometimes within introns of their target genes. Beyond that, some enhancers can bypass their nearest genes to regulate more distant ones along a chromosome. In some special cases, they can even regulate genes on another chromosome [2]. This kind of locational uncertainty has made their prediction a very challenging task for modern biologists.

The very early attempts to identify enhancers on a genome-wide scale started with biologically experimental methods. The most representative one is chromatin immunoprecipitation followed by deep sequencing (ChIP-seq) [3]. By targeting at enhancer associated marks like transcription activator p300, histone H3 monomethylated at K4 (H3K4me1) and H3 acetylated at lysine 27 (H3K27ac) [4-6], this technique has been successfully applied in some cell lines to identify enhancers. Another category of experimental method is based on chromatin accessibility, i.e., detecting the open regions on the genome. Two commonly used techniques are DNase I digestion coupled to sequencing (DNase-seq) [7] and transposase-accessible chromatin followed by sequencing (ATAC-seq) [8]. In addition, further studies on enhancers have shown that they can function as transcriptional units and produce non-coding RNAs (eRNAs), which are hallmarks of active enhancers [9]. But eRNAs are generally unstable and have a short half-life, which make them extremely hard to detect in cells. Despite of this, several techniques have been developed to detect the expression of eRNAs, like global run-on sequencing (GRO-seq) and cap-analysis of gene expression (CAGE) [10-11]. While all of the aforementioned experimental methods are useful to some extent, there is currently no golden standard in biology for enhancer identification. And beyond that, experimental ways are time-consuming and labor-intensive. It's basically impractical at present to identify enhancers for all the cell types at various stages.

On this account, some computational methods have been put forward to fill this gap. As the first attempt, Heintzman et al. developed a computational prediction algorithm to locate enhancers in the ENCODE regions of HeLa cells based on similarity to the training set chromatin profiles [5]. Firpi et al. further proposed a computational framework named CSI-ANN [12]. It was composed of a data transformation and a feature extraction step followed by a classification step with time-delay neural network. With the discovery and map of more and more histone modifications, the selection of the optimal set from the entire range of chromatin modifications for enhancer identification become an urgent question for biologists. So in the work of Rajagopals et al., they developed RFECS, a Random-Forest based algorithm to integrate 24 histone modification profiles in all for identification of enhancers in several cell types [13]. They claimed that their method not only led to more accurate predictions but also identified the most informative and robust set of three chromatin marks for enhancer identification. However, the goals of all the aforementioned methods were simply to label a DNA sequence as an enhancer or not. They neglected to determine the strength of enhancers, i.e., their activity level, which is also biologically meaningful. Given that, Liu et al. proposed a two-layer predictor called iEnhancer-2L by formulating DNA elements with pseudo k-tuple nucleotide composition [14]. In the second layer of their predictor, they identified the strength of enhancers for the very first time. Considering the poor performance of iEnhancer-2L, they further improved their algorithm by formulating sequences with different feature representations and using ensemble learning in their later work iEnhancer-EL [15]. On the basis of Liu et al.'s work, Jia et al. developed a predictor called EnhancerPred by extracting three types of sequence-based features and using support vector machine to identify enhancers and their strength [16]. Recently, Cai et al. proposed a more advanced predictor named iEnhancer-XG. In their method, as many as five different sequence derived feature representations were used as the input of XG-boost, a new learning algorithm based on gradient boosted decision trees [17].

Over the past few years, deep learning has seen a comprehensive penetration into various fields, including computer vision, natural language processing and even bioinformatics [18-23]. Naturally, a torrent of deep learning based methods for enhancer identification have sprung up, such as EP-DNN, BiRen, DECRES, DeepEnhancer and so on [24-27]. Comparing to traditional machine learning methods, deep learning obviates the need for manually curating features and can unearth informative hidden patterns in the data. While these deep learning based enhancer predictors give much better performance than that of traditional machine learning based ones, their weaknesses are also quite obvious, i.e., the prediction of enhancer strength is not reflected in their methodologies.

We here present a two-layer enhancer predictor named DENIES by utilizing DNA shape information besides two common sequence-derived features (i.e. one-hot encoding and *k*-mer) as the input of our developed deep learning model. DNA shape refers to the three-dimensional structures of DNA. While several studies have pointed out its extremely importance in biology, the appliance of this particular feature has been limited to the modeling of TF-DNA binding [28-29]. In the study of iEnhancer-2L, Liu et al. incorporated six DNA local structural parameters into their formulated pseudo *k*-tuple nucleotide composition [14]. But these local structural parameters are just predicted from the physicochemical properties of two neighboring base pairs and unable to accurately reflect the three-dimensional structure of DNA sequences. Given this, we use DNAshape [30], a Monte

Carlo (MC) simulation based method, to derive more accurate DNA shape features in our study. Comparing to existing state-of-the-art methods, the performance of our proposed predictor gets an obvious boost for both layers of enhancer identification. More importantly, it proves that DNA shape can be used as another major feature to enhance the identification of enhancers and their strength.

2. Materials and Methods

2.1. Benchmark and independent datasets

The benchmark and independent datasets used in our study were obtained from a series of previous works [14-17]. The construction of the two datasets was based on the chromatin state annotation. Specifically, Ernst et al. mapped nine chromatin marks across nine cell types and used recurrent combinations of these marks to define 15 chromatin states including repressed, poised and active promoters, strong and weak enhancers and so on. Then ChromHMM was developed using a multivariate Hidden Markov Model to learn these chromatin states information and got the genome-wide chromatin state annotation in these different cell lines [31-32]. After that, different sorts of DNA sequences were selected as candidate samples to construct the two datasets based on the genome-wide chromatin state annotation given by ChromHMM in a total of nine cell types.

A detailed description of the post processing on these candidate samples can be looked up in the original works of Liu et al [14-15]. Here, we just report the composition of the final datasets. The benchmark dataset consists of 2968 sequence samples, each of which is 200 bp long. Among these, 1484 samples are enhancers and the other 1484 samples are non-enhancers. They are used to construct the first layer predictor. Of the 1484 enhancer samples, strong enhancers and weak enhancers are half and half. They are used to construct the second layer predictor. The independent dataset was constructed based on the same protocol as used to construct the benchmark dataset. It's composed of 200 enhancers and 200 non-enhancers. Of the 200 enhancers, one half are strong enhancers, and the other half are weak enhancers. These samples of the independent dataset are used to evaluate the performance of predictors developed using the benchmark dataset. Note that there's no overlap between the samples in benchmark dataset and independent dataset [15].

2.2. Feature representation

A fundamental problem in developing a bioinformatics related predictor is how to formulate a biological sequence (i.e. DNA sequence in our study) with some specific feature representations. While an effective feature representation is able to extract the most informative patterns in a sequence, an unsatisfactory one can hardly ensure the integrality of the information. In truth, quite a lot of feature representations for biological sequences have been proposed so far, such as *k*-mer, pseudo *k*-tuple nucleotide composition (PseKNC), subsequence and mismatch profile [33]. In this study, we explore three feature representation methods, namely, one-hot, *k*-mer and DNA shape. Among these, one-hot and *k*-mer represent two common ways to formulate DNA sequences. They have been widely used in bioinformatics to resolve various types of classification and prediction problems. However, sequence-constraint methods often fail to identify non-coding functional elements like enhancers because they neglect to consider the three-dimensional structures of DNA sequences [34]. Hence, DNA shape is used as another feature representation method in our study for describing the 3D structures of DNA sequences. A detailed description of these three different feature representation methods is as below.

2.2.1. K-mer

Oligomers of length *k*, or *k*-mer refers to all the subsequences of length *k* contained within a biological sequence. It's a widely used and probably the simplest feature representation method for modeling the properties and functions of biological sequences [33]. In the case of deoxyribonucleic acid, every sequence is composed of four different types of nucleotides (i.e. A, C, G and T). *K*-mer approach will first list all the possible subsequences of length *k* and scan the whole DNA sequence to find the occurrence frequency of each subsequence. Then the occurrence frequency of each subsequence will be combined by the order of the listed subsequences. This combined feature vector is called the *k*-mer feature vector of that sequence. Suppose we have a sequence **S**, then the *k*-mer feature vector of **S** can be defined as:

$$f^s = [y_1^s,\ y_2^s,\ \cdots,\ y_i^s,\ \cdots,\ y_L^s] \tag{1}$$

where y_i^s is the occurrence frequency of the *i*th *k* neighboring nucleotides in the sequence **S** and the value of *L* is 4^k. As pointed in [35], the selection of the parameter *k* in *k*-mer feature representation is of great difficulty owing to its inherent limitation. The *k*-mer feature vector tends to get sparser and encode less efficient information when *k* gradually increases. We observed that the *k*-mer vector is quite sparse when the value of *k* is higher than 5 in our case since the length of enhancer sequences in our dataset is comparatively short. Here, we use a combination of different *k*-mer feature vectors where *k* ranges from 1 to 5. Then these vectors are concatenated to a final feature vector.

2.2.2. One-hot encoding

One-hot encoding is another common feature representation method for formulating DNA sequences. With deep learning gradually penetrated into bioinformatics, this encoding scheme is very popular when it's combined in use with convolutional neural networks (CNN) [20-25]. With this feature representation, every nucleotide in the sequence will be transformed into a four-dimensional binary vector with the bit marking the current nucleotide set to one and all the other bits set to zero. Specifically, the four nucleotides, A (adenine), C (cytosine), G (guanine), T (thymine) will be transformed into binary vectors $(1, 0, 0, 0)^{\mathrm{T}}$, $(0, 1, 0, 0)^{\mathrm{T}}$, $(0, 0, 1, 0)^{\mathrm{T}}$, $(0, 0, 0, 1)^{\mathrm{T}}$ respectively. Then the binary vector for each nucleotide in the DNA sequences will be merged into a binary matrix. Given a DNA sequence $\mathbf{S} = \mathrm{N}_1\,\mathrm{N}_2 \cdots \mathrm{N}_i \cdots \mathrm{N}_L$, then the one-hot encoding matrix **M** for sequence **S** can be formulated as:

$$\mathbf{M} = [f(N_1)\, f(N_2) \;\cdots\; f(N_i) \;\cdots\; f(N_L)] \tag{2}$$

where f is a function defined as:

$$f(N_i) = \begin{cases} (1,0,0,0)^{\mathrm{T}};\ if\ N_i = A \\ (0,1,0,0)^{\mathrm{T}};\ if\ N_i = C \\ (0,0,1,0)^{\mathrm{T}};\ if\ N_i = G \\ (0,0,0,1)^{\mathrm{T}};\ if\ N_i = T \end{cases} \tag{3}$$

2.2.3. DNA shape

DNA shape presents the chromatin structural information about DNA sequences with which the other two classic feature representation methods can not provide. As distal cis-regulatory elements that can activate downstream genes, the chromatin structures of enhancer regions tend to be open so as to provide a protein-binding platform for a combination of transcription factors and co-factors. In fact, several former studies have pointed out the role of DNA shape in the recognition of three-dimensional DNA structures for transcription factors [36-38]. Before long Zhou et al. proposed a high-throughput method called DNAshape for predicting chromatin structural features from DNA sequences [28]. The core of their methodology is a pentamer based model built from all-atom Monte Carlo simulations where a sliding-window approach is used to mine DNA shape features from DNA sequences. Later, Chiu et al. developed DNAshapeR, an R/Bioconductor package on the basis of DNAshape to generate DNA shape predictions in an ultra-fast, high-throughput and user-friendly manner [39]. At first, only four DNA shape features were included, that is minor groove width (MGW), helix twist (HelT), propeller twist (ProT) and Roll, for their extremely importance in the recognition of DNA structures. And in their latest package release, another 9 DNA shape features were added and the entire repertoire was finally expanded to a total of 13. Among these, seven features were nucleotide shape parameters and the other six were base pair-step parameters. A simple sketch on explaining the distinction between generating the two types of DNA shape parameters is presented in Fig. 1 where the DNA sequence is scanned with a pentamer sliding window. For each pentamer subsequence currently being scanned, a DNA shape prediction value of the central nucleotide or two prediction values of the two central base pair steps will be computed based on the specific type of the given DNA shape parameter. Supposing the length of a DNA sequence is N, then the dimension of nucleotide shape parameter-based feature vector and base pair-step shape parameter-based feature vector can be easily induced as N − 4 and N − 3 respectively. As there are 7 nucleotide shape parameters and 6 base pair-step shape parameters used in our study, the length of the concatenated shape feature vector can be formulated as $7 \times (N - 4) + 6 \times (N - 3)$. Given N = 200 in our case, an input DNA sequence can be finally encoded with a DNA shape vector of dimension 2554.

2.3 Network architecture

The detailed network architecture of our designed deep learning model is depicted in Fig. 2. The model is mainly composed of four modules that are already filled with a light blue background. The top three modules, from left, are DNA shape module, one-hot module and *k*-mer module respectively. A detailed description of each module is as follows. In DNA shape module, the R package DNAshapeR is used to generate the shape feature predictions from the given set of DNA shape parameters as the protocol illustrated in Fig. 1. In our study, all the 13 DNA shape features were generated for our enhancer dataset. In one-hot encoding module, the input sequence is firstly encoded with a 4 × 200 one-hot binary matrix and then fed into a two-layer convolution neural network. Sixteen convolutional kernels with dimension 4 × 8 and thirty-two convolutional kernels with dimension 1 × 8 are used for the first and second convolutional layer respectively. Notably, while it hasn't been shown in the figure for simplicity, each convolutional layer is followed by a max pooling layer with dimension 1 × 2. After convolutions, the output feature maps will be flattened to a feature vector. And in kmer module, the frequency vector of different k-mers (k ranges from 1 to 5) will be computed separately and then concatenated. Finally, output vectors of the top three modules are concatenated and sent into the joint module where a multilayer perceptron network (MLP) with two hidden layers is used. The number of neurons used for the first and second layers are 512 and 64 respectively. The output layer has only one neuron representing the binary classification result of our network.

2.4. Configuration and implementation

All the models built in our study are implemented with Pytorch 1.4.0 [40]. Adam [41] is chosen as our optimizer algorithm and the learning rate for training models is set to 1e-5. The mini-batch size is 20 and binary cross entropy is employed as the loss function. To prevent models from overfitting, the early stopping strategy is used with patience set to 30 and the evaluation metric MCC is monitored in validation set. That is to say, after a successive of 30 training epochs with no increase on the metric MCC, the training process is stopped and the model at the epoch with the highest MCC value will be saved for evaluation on the independent dataset.

2.5. Performance evaluation

For a fair comparison, the identical set of evaluation metrics as used in a series of previous works is also adopted in our study. These metrics are sensitivity (SN), specificity (SP), accuracy (ACC), and Matthew correlation coefficient (MCC). They are all formulated as below:

$$\begin{cases} SN = \dfrac{TP}{TP + FN} \\ SP = \dfrac{TN}{TN + FP} \\ ACC = \dfrac{TP + TN}{TP + TN + FP + FN} \\ MCC = \dfrac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \end{cases} \tag{4}$$

where TP, TN, FP, FN represent the number of true positives, true negatives, false positives and false negatives respectively. Considering our dataset is comparatively small, a five-fold cross validation is used to evaluate the performance of our model. Specifically, our benchmark dataset is partitioned into five folds. Every time four parts of them will be used as the training set and the remaining one will be used as the validation set. This process is repeated after 5 times and each time we will get a different data partition for the training of our model. Then the five trained models will be tested in turn on the independent dataset. As there are a total of five predicts given by the respective trained models, we further adopt an ensemble learning strategy to get the final predict on the independent dataset (i.e. taking the mean value of the five predicts). To reflect the stability of our models, the five-fold cross validation experiment is conducted for ten times in all and the results are shown with box plots.

3. Results

3.1. Performance comparison between the basic set and full set of DNA shape features

DNAshape initially provided the prediction of only 4 DNA shape features, namely MGW, ProT, Roll and HelT, which we refer to as the basic set. Another 9 features (Rise, Roll, Shift, Slide, Tilt, Buckle, Opening, Shear, Stagger and Stretch) were added in the latest version of DNAshapeR and the feature set was expanded to a total of 13, which we refer to as the full set. While former studies focused mainly on the basic set, the full set of DNA shape features were used in our study. Naturally, it’s worth evaluating whether these 9 new added DNA shape features have a positive effect on the identification of enhancers and their strength. To prove that, we compared the performance of basic set and full set of DNA shape features on the independent dataset. The results of the performance evaluation on five metrics have been shown in Fig. 2. For the first layer targeting at distinguishing enhancers from non-enhancers, the performance of full set is better than that of basic set by a narrow margin on four evaluation metrics except for sensitivity (SN). While for the second layer aiming at distinguishing strong enhancers from weak enhancers, the edge of full set comparing to basic set is more obvious with a lead of performance on all five evaluation metrics. Obviously, the introduction of the additional 9 DNA shape features has a positive influence for both layers of enhancer prediction, especially for the second layer.

3.2. Visualization of DNA shape features

Besides the prediction of DNA shape features, another outstanding function provided by DNAshapeR is its graphical representation, which means various shape features predicted from DNA sequences can be visualized for further analysis. Here, we use the function plotShape given in the software package to visualize DNA shape features as aggregated line plots (i.e., the shape feature values of all the sequence samples are aggregated in column direction to get the mean value). The shape features of positive set and negative set are both shown in a single picture to reflect the difference. The line plots of some representative shape features have been chosen and shown in Fig. 4 and Fig. 5. For minor groove width, the two aggregated lines are separated on the first layer while overlapped on the second layer, which suggests that MGW is an efficient shape feature to distinguish enhancers from non-enhancers but may not be ideal for distinguishing strong enhancers from weak enhancers. For the DNA shape feature Opening, the two aggregated lines are obviously separated on both layers suggesting its effectiveness for both layers of enhancer identification. For Buckle, though the aggregated line of positive set is higher than that of negative set, there's still some overlap between them. And for Tilt, the profiles of both lines just fluctuate around zero, which may suggest this shape feature can hardly be used for enhancer identification task.

3.3. Performance of different feature representations and their combinations

As there are total three different types of feature representations used in our method, a natural question in face of us is to determine the importance of them in the identification of enhancers. For that, each module used in Fig. 2 is taken out and designed as an individual network. Besides, different modules are combined to see the effect of various combinations of these feature representations. For the sake of simplicity, the model combining one-hot and DNA shape modules is denoted as 'O-S'; the model combining one-hot and k-mer modules is denoted as 'O-K'; the model combining DNA shape and k-mer modules is denoted as 'S-K'; the model combining all three modules is denoted simply as 'ALL'. Then the performance of all these models is evaluated with a five-fold cross validation on the independent dataset. To reflect the stability of these models, the cross validation experiment is repeated after 10 times. The results of the ten experiments have been shown with box plot in Fig. 6 and the mean values of the five evaluation metrics are given in Table 1 and Table 2. For the first layer, kmer approach achieves the best performance among the three single feature representations with ACC at 75.8% and MCC at 0.516. The performance of the other two feature representations, namely one-hot and shape, are very close with accuracy over 73.5% and MCC around 0.47. While for the second layer prediction, the performance of one-hot feature is much unsatisfactory with ACC at 57.95% and MCC at 0.163. For the other two feature representations, the performance of DNA shape is slightly lower than that of kmer. Their ACC and MCC are over 62% and 0.24 respectively. Overall, kmer outperforms the other two feature representations on both layers. The performance of DNA shape comes next and one-hot performs worst among the three feature representations. As for those combined feature representations, their performance basically all improved compared to that of single feature representations. Notably, the model combining all three feature representations performed best on both layers. For the first layer, it achieves the best performance on all evaluation metrics except for sensitivity (SN). And for the second layer, it outperforms the other models on all evaluation metrics except for specificity (SP). For that reason, it's selected as our final model for performance comparison with a series of state-of-the-art methods.

3.4. Performance comparison with existing methods

We compare the performance of our final model on the independent dataset with some existing state-of-the-art works. The result of comparison has been listed in Table 3. It can be seen from the table that our method achieves the highest performance on most evaluation metrics for both layers. Note that ACC and MCC are deemed as the two most important ones among the five evaluation metrics for our prediction task [15]. For the first layer, ACC is improved by over one percentage point and MCC is improved by 0.024. And for the second layer, these two metrics are boosted by 4.75% and 0.108. As another important evaluation metric for binary classification, AUC is improved from 0.817 to 0.834 and from 0.680 to 0.753 for the first and second layer respectively. On the whole, the performance of our method surpasses these existing methods terms of a comprehensive comparison on these evaluation metrics.

4. Discussion and conclusion

As the core parts of DNA regulatory elements responsible for regulating gene expression, enhancers have always been paid the most attention in bioinformatics. However, their locational uncertainty and the poor understanding of their sequence code have made their identification an extremely challenging task [2]. To further enhance the identification of enhancers and their strength, most of the current machine learning based predictors tend to derive more feature representations from DNA sequences and ensemble the prediction results of individual features. While some of the existing structural biology and genomics studies have confirmed the relationship between TF-DNA binding and the recognition of chromatin structures, we observe that the sequence-derived feature representations used in previous works cannot reflect the structural information of DNA sequences. In light of this, DNA shape is used as an additional feature input besides two commonly used ones, i.e., one-hot encoding and kmer for our developed deep learning model. Through the ablation experiment, we find that the performance of feature combined models is boosted comparing to those single feature representation based ones. Above all, the deep learning model with all the three feature representations as input achieves best performance on both layers of enhancer identification. And in the comparison with some state-of-the-art methods, DENIES achieves remarkable performance with only three features all derived from DNA sequences.

Yet, our study can be further improved from two main aspects. One is to increase the interpretability of our method. An inherent and obvious drawback of deep learning based method is its poor interpretability since it operates like a black box. In our method, we throw ten more DNA shape features into the deep learning model for training, but the importance degree of each feature in the identification of enhancers is unclear for us. Though we visualize different DNA shape features as aggregated line plots, there still needs a method to quantitatively determine the importance degrees of them. In the work of RFECS [13], they developed a new type of random forest algorithm named vector-random forest by utilizing linear discriminant at each node. The most significant feature of this random forest is that the nodes of decision trees can be vectors. By utilizing this new type of random forest algorithm, each DNA shape feature will be given a score representing its importance degree in identifying enhancers. We can further rank these DNA shape features by the importance scores to derive an optimal set of DNA shape features for identifying enhancers. Another aspect is to integrate epigenetic features like histone modifications. Many studies have associated the enhancer activities with certain characteristic histone modification patterns [5, 13, 42]. However, histone marks are cell line specific while the enhancer dataset used in our study are not, so it's beyond our reach at present. Nevertheless, it doesn't affect us to apply this method in later constructed cell line specific enhancer dataset.

References

1. Maston, G. A., Evans, S. K., & Green, M. R. (2006). Transcriptional regulatory elements in the human genome. Annual review of genomics and human genetics, 7, 29–59.
2. Pennacchio, L. A., Bickmore, W., Dean, A., Nobrega, M. A., & Bejerano, G. (2013). Enhancers: five essential questions. Nature reviews. Genetics, 14(4), 288–295.
3. Mardis E. R. (2007). ChIP-seq: welcome to the new frontier. Nature methods, 4(8), 613–614.
4. Visel, A., Blow, M. J., Li, Z., Zhang, T., Akiyama, J. A., Holt, A., Plajzer-Frick, I., Shoukry, M., Wright, C., Chen, F., Afzal, V., Ren, B., Rubin, E. M., & Pennacchio, L. A.

(2009). ChIP-seq accurately predicts tissue-specific activity of enhancers. Nature, 457(7231), 854–858.

5. Heintzman ND, Stuart RK, Hon G, Fu Y, Ching CW, Hawkins RD, Barrera LO, Van Calcar S, Qu C, Ching KA, Wang W, Weng Z, Green RD, Crawford GE, Ren B. Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. Nat Genet. 2007 Mar;39(3):311-8.
6. Creyghton, M. P., Cheng, A. W., Welstead, G. G., Kooistra, T., Carey, B. W., Steine, E. J., Hanna, J., Lodato, M. A., Frampton, G. M., Sharp, P. A., Boyer, L. A., Young, R. A., & Jaenisch, R. (2010). Histone H3K27ac separates active from poised enhancers and predicts developmental state. Proceedings of the National Academy of Sciences of the United States of America, 107(50), 21931–21936.
7. Dorschner, M. O., Hawrylycz, M., Humbert, R., Wallace, J. C., Shafer, A., Kawamoto, J., Mack, J., Hall, R., Goldy, J., Sabo, P. J., Kohli, A., Li, Q., McArthur, M., & Stamatoyannopoulos, J. A. (2004). High-throughput localization of functional elements by quantitative chromatin profiling. Nature methods, 1(3), 219–225.
8. Buenrostro, J. D., Wu, B., Chang, H. Y., & Greenleaf, W. J. (2015). ATAC-seq: A Method for Assaying Chromatin Accessibility Genome-Wide. Current protocols in molecular biology, 109, 21.29.1–21.29.9.
9. Kim, T. K., Hemberg, M., Gray, J. M., Costa, A. M., Bear, D. M., Wu, J., Harmin, D. A., Laptewicz, M., Barbara-Haley, K., Kuersten, S., Markenscoff-Papadimitriou, E., Kuhl, D., Bito, H., Worley, P. F., Kreiman, G., & Greenberg, M. E. (2010). Widespread transcription at neuronal activity-regulated enhancers. Nature, 465(7295), 182–187.
10. Core, L. J., Waterfall, J. J., & Lis, J. T. (2008). Nascent RNA sequencing reveals widespread pausing and divergent initiation at human promoters. Science (New York, N.Y.), 322(5909), 1845–1848.
11. Kodzius, R., Kojima, M., Nishiyori, H., Nakamura, M., Fukuda, S., Tagami, M., Sasaki, D., Imamura, K., Kai, C., Harbers, M., Hayashizaki, Y., & Carninci, P. (2006). CAGE: cap analysis of gene expression. Nature methods, 3(3), 211–222.
12. Firpi, H. A., Ucar, D., & Tan, K. (2010). Discover regulatory DNA elements using chromatin signatures and artificial neural network. Bioinformatics (Oxford, England), 26(13), 1579–1586.
13. Rajagopal, N., Xie, W., Li, Y., Wagner, U., Wang, W., Stamatoyannopoulos, J., Ernst, J., Kellis, M., & Ren, B. (2013). RFECS: a random-forest based algorithm for enhancer identification from chromatin state. PLoS computational biology, 9(3), e1002968.
14. Liu, B., Fang, L., Long, R., Lan, X., & Chou, K. C. (2016). iEnhancer-2L: a two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition. Bioinformatics (Oxford, England), 32(3), 362–369.
15. Liu, B., Li, K., Huang, D. S., & Chou, K. C. (2018). iEnhancer-EL: identifying enhancers and their strength with ensemble learning approach. Bioinformatics (Oxford, England),

34(22), 3835–3842.

16. Jia, C., & He, W. (2016). EnhancerPred: a predictor for discovering enhancers based on the combination and selection of multiple features. Scientific reports, 6, 38741.
17. Cai, L., Ren, X., Fu, X., Peng, L., Gao, M., & Zeng, X. (2020). iEnhancer-XG: Interpretable sequence-based enhancers and their strength predictor. Bioinformatics (Oxford, England), btaa914. Advance online publication.
18. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
19. Collobert, R., & Weston, J. (2008, July). A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning (pp. 160-167).
20. Zhou, J., & Troyanskaya, O. G. (2015). Predicting effects of noncoding variants with deep learning-based sequence model. Nature methods, 12(10), 931–934.
21. Alipanahi, B., Delong, A., Weirauch, M. T., & Frey, B. J. (2015). Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nature biotechnology, 33(8), 831–838.
22. Quang, D., & Xie, X. (2016). DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. Nucleic acids research, 44(11), e107.
23. Kelley, D. R., Snoek, J., & Rinn, J. L. (2016). Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome research, 26(7), 990–999.
24. Yang, B., Liu, F., Ren, C., Ouyang, Z., Xie, Z., Bo, X., & Shu, W. (2017). BiRen: predicting enhancers with a deep-learning-based model using the DNA sequence alone. Bioinformatics (Oxford, England), 33(13), 1930–1936.
25. Min, X., Zeng, W., Chen, S., Chen, N., Chen, T., & Jiang, R. (2017). Predicting enhancers with deep convolutional neural networks. BMC bioinformatics, 18(Suppl 13), 478.
26. Li, Y., Shi, W., & Wasserman, W. W. (2018). Genome-wide prediction of cis-regulatory regions using supervised deep learning methods. BMC bioinformatics, 19(1), 202.
27. Liu, F., Li, H., Ren, C., Bo, X., & Shu, W. (2016). PEDLA: predicting enhancers with a deep learning-based algorithmic framework. Scientific reports, 6, 28517.
28. Zhou, T., Shen, N., Yang, L., Abe, N., Horton, J., Mann, R. S., Bussemaker, H. J., Gordân, R., & Rohs, R. (2015). Quantitative modeling of transcription factor binding specificities using DNA shape. Proceedings of the National Academy of Sciences of the United States of America, 112(15), 4654–4659.
29. Zhang, Q., Shen, Z., & Huang, D. S. (2021). Predicting in-vitro Transcription Factor Binding Sites Using DNA Sequence + Shape. IEEE/ACM transactions on computational biology and bioinformatics, 18(2), 667–676.

30. Zhou, T., Yang, L., Lu, Y., Dror, I., Dantas Machado, A. C., Ghane, T., Di Felice, R., & Rohs, R. (2013). DNAshape: a method for the high-throughput prediction of DNA structural features on a genomic scale. Nucleic acids research, 41(Web Server issue), W56–W62.
31. Ernst, J., Kheradpour, P., Mikkelsen, T. S., Shoresh, N., Ward, L. D., Epstein, C. B., Zhang, X., Wang, L., Issner, R., Coyne, M., Ku, M., Durham, T., Kellis, M., & Bernstein, B. E. (2011). Mapping and analysis of chromatin state dynamics in nine human cell types. Nature, 473(7345), 43–49.
32. Ernst, J., & Kellis, M. (2012). ChromHMM: automating chromatin-state discovery and characterization. Nature methods, 9(3), 215–216.
33. Liu B. (2019). BioSeq-Analysis: a platform for DNA, RNA and protein sequence analysis based on machine learning approaches. Briefings in bioinformatics, 20(4), 1280–1294.
34. Friedel, M., Nikolajewa, S., Sühnel, J., & Wilhelm, T. (2009). DiProGB: the dinucleotide properties genome browser. Bioinformatics (Oxford, England), 25(19), 2603–2604.
35. Ghandi, M., Lee, D., Mohammad-Noori, M., & Beer, M. A. (2014). Enhanced regulatory sequence prediction using gapped k-mer features. PLoS computational biology, 10(7), e1003711.
36. Joshi, R., Passner, J. M., Rohs, R., Jain, R., Sosinsky, A., Crickmore, M. A., Jacob, V., Aggarwal, A. K., Honig, B., & Mann, R. S. (2007). Functional specificity of a Hox protein mediated by the recognition of minor groove structure. Cell, 131(3), 530–543.
37. Rohs, R., West, S. M., Sosinsky, A., Liu, P., Mann, R. S., & Honig, B. (2009). The role of DNA shape in protein-DNA recognition. Nature, 461(7268), 1248–1253.
38. White, M. A., Myers, C. A., Corbo, J. C., & Cohen, B. A. (2013). Massively parallel in vivo enhancer assay reveals that highly local features determine the cis-regulatory function of ChIP-seq peaks. Proceedings of the National Academy of Sciences of the United States of America, 110(29), 11952–11957.
39. Chiu, T. P., Comoglio, F., Zhou, T., Yang, L., Paro, R., & Rohs, R. (2016). DNAshapeR: an R/Bioconductor package for DNA shape prediction and feature encoding. Bioinformatics (Oxford, England), 32(8), 1211–1213.
40. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703.
41. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
42. Hon, G., Ren, B., & Wang, W. (2008). ChromaSig: a probabilistic approach to finding common chromatin signatures in the human genome. PLoS computational biology, 4(10), e1000201.

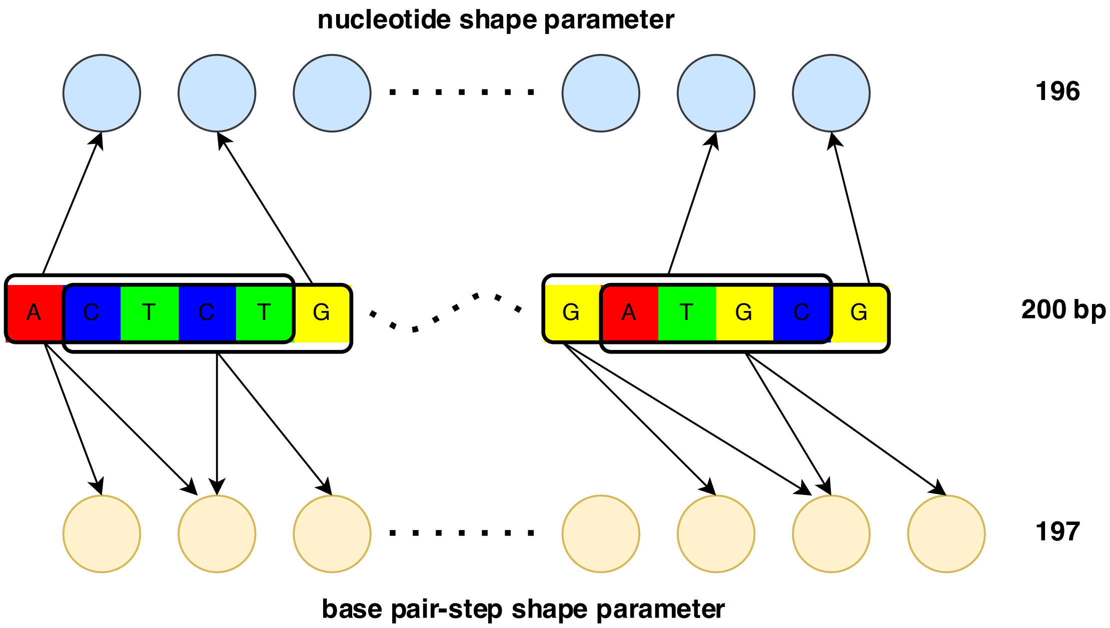


Figure 1. A simple sketch on illustrating the difference between the generation of nucleotide shape parameters and base pair-step shape parameters. The DNA sequence is scanned with a pentamer sliding window to derive DNA shape feature vectors. For each pentamer subsequence being scanned, a single prediction value of the central nucleotide will be computed for nucleotide parameters like MGW and ProT. While for base pair-step shape parameters like Roll and HelT, the prediction values of the two central base pair steps will be provided. Take the first pentamer in the sequence as an example, the central nucleotide is T and the two central base pair steps are CT and TC respectively. It's worth noting that the second central base pair step of a pentamer subsequence is identical to the first central base pair step of the next pentamer subsequence, so they share the same DNA shape prediction value.

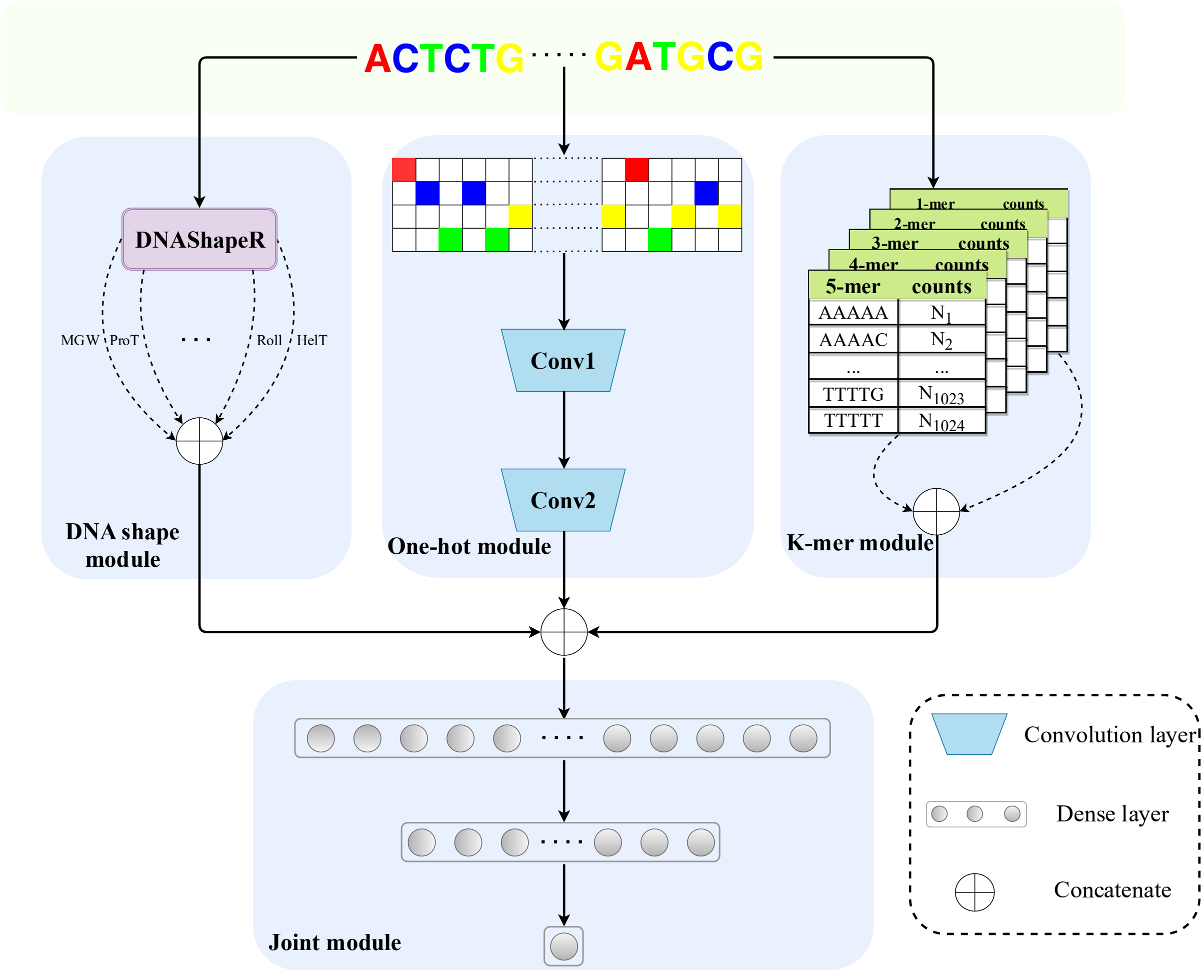


Figure 2. The network architecture of our designed deep learning model. The top three modules, from left, are DNA shape module, one-hot module and kmer module respectively and the bottom one is the joint module. The output vector from the top three modules will be concatenated and fed into the joint module where a multilayer perceptron (MLP) is used to get the final prediction result.

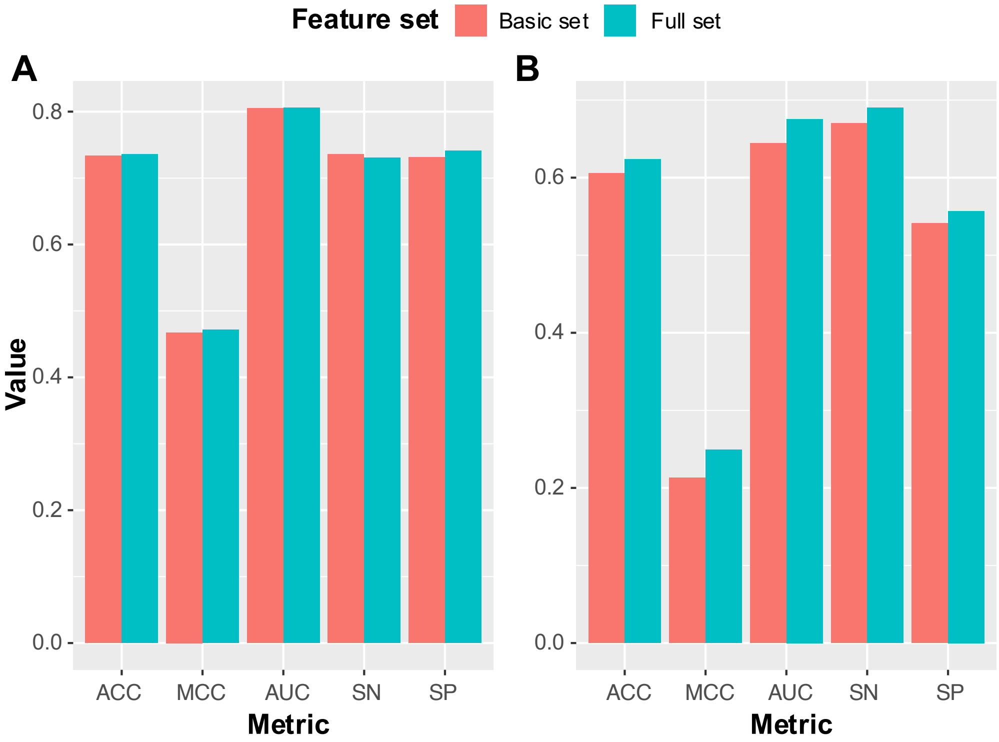


Figure 3. Performance comparison between the basic set and full set of DNA shape features on the (**A**) first layer and (**B**) second layer.

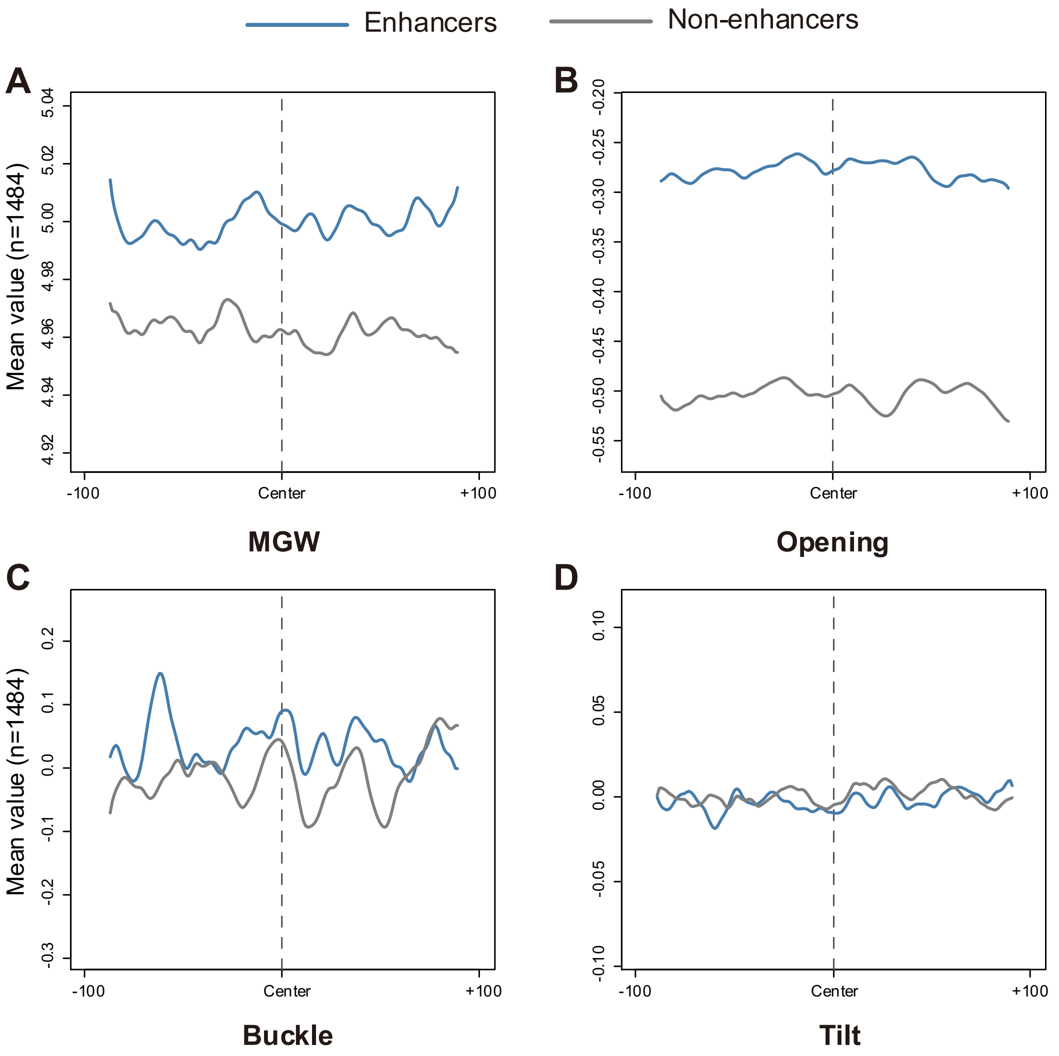


Figure 4. Visualization of four representative DNA shape feature features (**A**) MGW (**B**) Opening (**C**) Buckle (**D**) Tilt with aggregated line plots on the first layer.

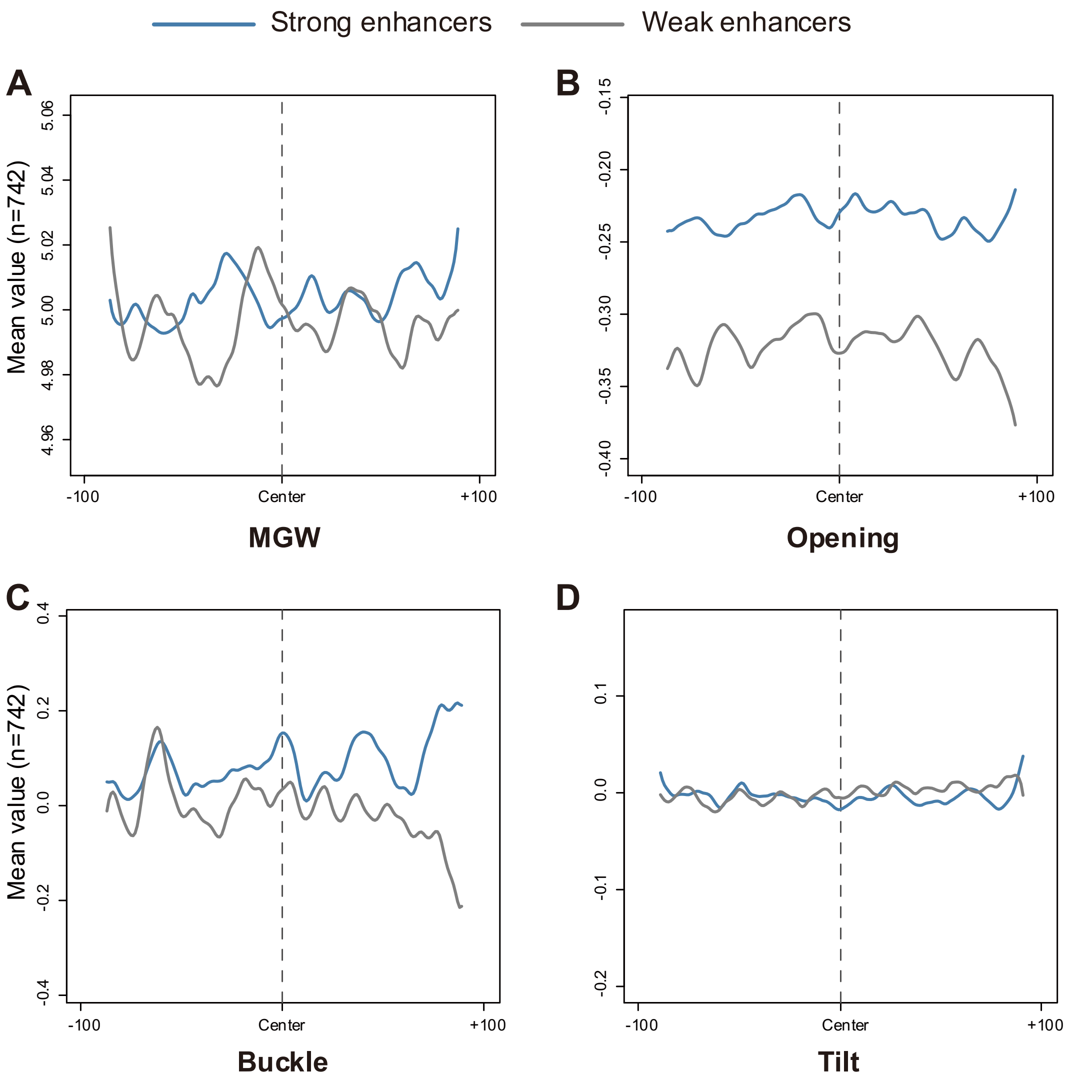
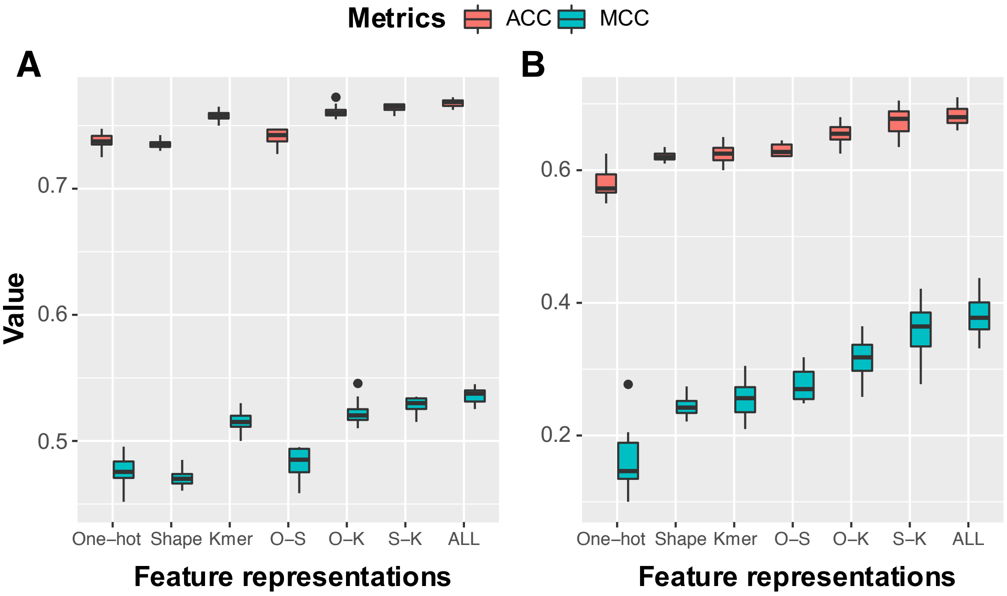


Figure 5. Visualization of four representative DNA shape feature features (**A**) MGW (**B**) Opening (**C**) Buckle (**D**) Tilt with aggregated line plots on the second layer.



554

555 Figure 6. Performance of different feature representations and their combinations.

556

557

Table 1 Performance of different feature representations and their combinations on the independent dataset of the first layer.

Feature representations	ACC (%)	MCC	AUC	SN (%)	SP (%)
One-hot	73.70	0.475	0.812	71.20	76.20
Shape	73.55	0.471	0.806	74.00	73.10
K-mer	75.80	0.516	0.830	76.00	75.60
O-S	74.13	0.483	0.814	72.85	75.40
O-K	76.15	0.523	0.834	76.25	76.05
S-K	76.40	0.528	0.833	**77.25**	75.55
All	**76.80**	**0.536**	**0.834**	76.90	**76.70**

The highest value achieved on every single metric has already been marked in bold.

Table 2 Performance of different feature representations and their combinations on the independent dataset of the second layer

Feature representations	ACC (%)	MCC	AUC	SN (%)	SP (%)
One-hot	57.95	0.163	0.638	66.40	49.50
Shape	62.05	0.243	0.679	68.40	**55.70**
K-mer	62.50	0.256	0.681	73.30	51.80
O-S	63.05	0.276	0.661	78.90	47.20
O-K	65.45	0.316	0.712	75.50	55.40
S-K	67.35	0.358	0.739	79.20	55.50
All	**68.25**	**0.380**	**0.753**	**82.00**	54.50

The highest value achieved on every single metric has already been marked in bold.

Table 3 Performance comparison with existing methods on the independent dataset

Task	Method	ACC (%)	MCC	AUC	SN (%)	SP (%)
1st layer	iEnhancer-2L	73.00	0.460	0.806	71.00	75.00
	Enhancer-Pred	74.00	0.480	0.801	73.50	74.50
	iEnhancer-EL	74.75	0.496	0.817	71.00	**78.50**
	iEnhancer-XG	75.75	0.515	-	74.00	77.50
	DENIES	**76.80**	**0.536**	**0.834**	**76.90**	76.70
2nd layer	iEnhancer-2L	60.50	0.218	0.668	47.00	**74.00**
	Enhancer-Pred	55.00	0.102	0.579	45.00	65.00
	iEnhancer-EL	61.00	0.222	0.680	54.00	68.00
	iEnhancer-XG	63.50	0.272	-	70.00	57.00
	DENIES	**68.25**	**0.380**	**0.753**	**82.00**	54.50

The highest value achieved on every single metric has already been marked in bold.