iEnhancer-5Step: Identifying enhancers using hidden information of DNA sequences via Chou's 5-step rule and word embedding

Nguyen Quoc Khanh Le, Edward Kien Yee Yapp, Quang-Thai Ho, N. Nagasundaram, Yu-Yen Ou, Hui-Yuan Yeh

Medical Humanities Research Cluster, School of Humanities, Nanyang Technological University, 48 Nanyang Ave, 639798, Singapore
Department of Computer Science and Engineering, Yuan Ze University, 32003, Taiwan
Singapore Institute of Manufacturing Technology, 2 Fusionopolis Way, #08-04, Innovis, 138634, Singapore

ARTICLE INFO

Keywords:
- Skip gram
- Continuous bag of words
- Regulatory transcription factor
- Support vector machine
- Two-layer classification
- Sequence analysis

ABSTRACT

An enhancer is a short (50–1500bp) region of DNA that plays an important role in gene expression and the production of RNA and proteins. Genetic variation in enhancers has been linked to many human diseases, such as cancer, disorder or inflammatory bowel disease. Due to the importance of enhancers in genomics, the classification of enhancers has become a popular area of research in computational biology. Despite the few computational tools employed to address this problem, their resulting performance still requires improvements. In this study, we treat enhancers by the word embeddings, including sub-word information of its biological words, which then serve as features to be fed into a support vector machine algorithm to classify them. We present iEnhancer-5Step, a web server containing two-layer classifiers to identify enhancers and their strength. We are able to attain an independent test accuracy of 79% and 63.5% in the two layers, respectively. Compared to current predictors on the same dataset, our proposed method is able to yield superior performance as compared to the other methods. Moreover, this study provides a basis for further research that can enrich the field of applying natural language processing techniques in biological sequences. iEnhancer-5Step is freely accessible via http://biologydeep.com/fastenc/.

1. Introduction

In genetics, an enhancer is a short (50–1500bp) region of DNA that plays an important role in gene expression and the production of RNA and proteins [1,2]. Enhancers are cis-acting and these RNAs are frequently alluded to as transcription factors. They can be situated at a distance of up to 1 Mbp (1,000,000bp) away from a gene, or even existing in different chromosomes, upstream or downstream from the transcription start site [2,3]. A huge number of enhancers are found in both eukaryotes and prokaryotes, especially in the human genome [4]. Genetic variation in enhancers has been linked to many human diseases, especially cancer [5,6], disorder [6,7], or inflammatory bowel disease [8].

Due to the importance of enhancers in genomics, the classification of the enhancers has become a popular area of study, especially in biological research. It has attracted many researchers from different areas of biology, such as pure biology, system biology, and computational biology. Lai et al. [9] have identified enhancers in the red flour beetle, Tribolium castaneum. Comparative genomics is also a good solution and it has been proven successful by Ref. [10]. Moreover, a new approach to this matter has been introduced by Zacher et al. [11] through the use of GenoSTAN to accurately identify the enhancer and promoter in roadmap epigenomics cell types. Furthermore, silico identification of enhancers has also been conducted based on a combination of transcription factor binding motif occurrences [12].

In bioinformatics, several computational methods have been adopted to identify enhancers from other regulatory elements. For instance, Firpi et al. [13] incorporated chromatin signatures and artificial neural network to discover enhancers. Erwin et al. [14] integrated a diversity of datasets to improve the identification of enhancers. Support vector machine (SVM) has been also used to predict mammalian enhancers [15]. Later, some of the improvements were done by using different machine learning and deep learning techniques, such as random forest [16], deep belief network [17], and deep-learning-based hybrid architecture [18]. In order to classify enhancers as strong and weak elements, there are a few predictors to this issue that should be...
looked at. The first study is iEnhancer-2L [19], in which a benchmark dataset to identify and classify enhancers was being provided. Therefore, they performed the experiments with pseudo-k-tuple nucleotide composition and reached an acceptable performance. There have been a few studies that used the same data set in their approach to increase the performance of this problem, such as using multiple features selection [20] and ensemble approach [21].

However, their performance results require much improvement, and in this study, we aim to enhance the solution to this issue with an innovative approach. Our idea is to transform the enhancer sequences into vectors using word embedding and then proceed to classify them with the use of effective neural networks. This idea has been indeed used in past experiments, where researches would attempt to apply existing natural language processing (NLP) algorithms to the study of biological sequences. It was first presented by Ref. [22] and applied successfully in many latter biological applications [23–25]. Moreover, the word feature namely k-mer has also been applied in RNA sequence description [26] and protein structure [27]. Even with enhancer sequences, NLP has been successfully applied in enhancer-promoter interactions [18, 28]. However, most researchers used the Word2Vec model, which each word in corpus like an atomic entity and generated a vector for each word. In this sense, Word2Vec is very much like Glove – both treating a word as the smallest unit to be trained. Here we present a new approach to transform DNA sequences into vectors by using the FastText model. FastText is an extension of Word2Vec proposed by Facebook in 2016, in which it treats each word as being composed of character n-grams [29]. Therefore, the vector for a word is made up of the sum of its character n-grams. FastText is shown to be more accurate than Word2Vec vectors in many fields [30, 31], and now we extend it into biological sequences.

Following this train of thought, we documented several key contributions of our study to the field of biology: (1) a new computational model for enhancers in which it is able to exhibit significant improvements beyond that of the previous predictors, (2) a new framework constructed from word embedding and supervised learning for classification of the DNA sequences with high performance, (3) a study that would provide much information to biologists and researchers as they are better equipped with the understanding of the enhancers, and (4) a basis for further study to apply natural language processing model in biological sequences. Here, we describe our contributions systematically.

As shown in a series of recent publications [32–38], to develop a really useful sequence-based statistical predictor for a biological or biomedical system, one should observe the guidelines of the 5-step rule [39]. In order to make the sequence of events clear, it follows as such: (i) how to construct or select a valid benchmark dataset to train and test the predictor; (ii) how to formulate the biological sequence samples with an effective mathematical expression that can truly reflect their intrinsic correlation with the target to be predicted; (iii) how to introduce or develop a powerful algorithm (or engine) to operate the prediction; (iv) how to properly perform cross-validation tests to objectively evaluate the anticipated accuracy of the predictor; (v) how to establish a user-friendly web-server for the predictor that is accessible to the public. Below, we would describe in detail how to carry out with these steps individually.

2. Materials and methods

We propose a novel approach through the use of word embedding and Support Vector Machine (SVM) [40] to classify enhancers with high performance. Fig. 1 illustrates a flowchart of the study, which is made up of two major processes: training skip-gram model (via the FastText library [29]) and supervised learning model (via SVM). The detailed description of the proposed approach would be discussed in the following paragraphs.

![Flowchart of this study](image)

**Fig. 1. Flowchart of this study.** First, we used benchmark dataset to train a FastText model and then used this model to generate vectors for DNA sequences. These vectors were then fed to SVM algorithm to perform a supervised learning.

### 2.1. Benchmark dataset

For this study, we re-used the benchmark dataset from the previous study [19]. This dataset was used in all enhancer classification problems, such as [19–21]. In this dataset, enhancers were collected according to the information of nine different cell lines and DNA sequences were extracted into fragments of 200 bp. CD-HIT [41] was also used to remove the pairwise sequences which had the similarity of more than 20%. The benchmark dataset includes 1,484 enhancers and 1,484 non-enhancers. 1,484 enhancers contain 742 strong enhancer samples and 742 weak enhancer samples to create the second layer classification. This study would also provide an independent dataset which would contain 200 non-enhancers and 200 enhancers (including 100 strong and 100 weak enhancers).

Thereafter, we randomly divided the training dataset into five separate sets to perform five-fold cross-validation. We performed the training five times, each time using a part set as a validation file and the other four files into a training set. The result of the cross-validation is an average value of five times training. Subsequently, the independent dataset was used to evaluate our model.

### 2.2. Novel distributed representation of DNA sequences

With an explosive growth of biological sequences in the post-genomic era, one of the most important but also the most difficult problem in computational biology is how to express a biological sequence with a discrete model or a vector, yet still retain considerable sequence-order information or key pattern characteristic. This is because all the existing machine-learning algorithms (such as “Optimization” algorithm [42], “Covariance Discriminant” or “CD” algorithm [43, 44], “Nearest Neighbor” or “NN” algorithm [45], and “Support Vector Machine” or “SVM” algorithm [45, 46]) can only handle vectors as elaborated in a comprehensive review [47]. However, a vector defined in a discrete model may completely lose all the sequence-pattern information. In order to avoid completely losing the sequence-pattern information for proteins, the pseudo amino acid composition [48] or PseAAC [49] was proposed. Ever since the concept
of Chou's PseaAC was proposed, it has been widely used in nearly all areas of computational proteomics (see, e.g. Refs. [50–58], as well as a long list of references cited in Ref. [59]). Recently, due to its widespread and increment of usage, three powerful open access softwares, called 'PseaAC-Builder' [60], 'propy' [61], and 'PseaAAC-General' [62], were established. The former two are for generating various modes of Chou's special PseaAC [63]. The 3rd one was created for Chou's general PseaAC [39], including not only all the special modes of feature vectors for proteins, but also the higher level feature vectors such as “Functional Domain” mode (see Eqs.9–10 of [39]), “Gene Ontology” mode (see Eqs.11–12 of [39]), and “Sequential Evolution” or “PSSM” mode (see Eqs.13–14 of [39]). Encouraged by the successes of using PseaAC to deal with protein/peptide sequences, the concept of PsekNC (Pseudo K-tuple Nucleotide Composition) [64] was developed for generating various feature vectors for DNA/RNA sequences [65,66] that have proved very useful as well. Particularly, a very recent yet powerful web-server called 'Pse-in-One' [67] and its updated version 'Pse-in-One2.0' [68] have been established that can be used to generate any desired pseudo components feature vectors for protein/peptide and DNA/RNA sequences according to the need of the users’ studies.

In this study, we applied a new feature extraction called word embedding representation which helped us interpret the hidden information of DNA sequences. In the NLP field, distributed representation has been successfully used to train word embedding and mapping of words to real-value vector space. In general, distributed representation characterizes an item through its relationship with alternative ones. In word embedding, similar words are mapped within the distributed representation vector space close to each other.

Inspired by the achievements of word embedding, which has been used in many NLP tasks, we treated our DNA sequences like natural language phrases. To perform this task, we divide the enhancers into a sequence of words with one word being one nucleobase (A, C, G, or T), two continuous nucleobases, and so on. For each DNA word, we then generated the word embedding using the language model. The purpose of this step is to encode the nucleobases by expressing their vector space distribution, thereby allowing it to be used for supervised learning algorithms. For the end-to-end machine learning, we proposed a supervised manner to learn the encoding directly from the raw nucleobase sequence in an unsupervised manner.

Due to the relevance between the natural language and biological sequence, many bioinformatics researchers have already been applying word embedding representation into biological sequences. For instance, it has been applied to deep proteomics and genomics [22], the identification of antimicrobial peptides [24], and HLA class I binding [69]. However, in all of these research experiments, Word2vec has been used to represent the biological sequences which resulted in certain limitations such as, not taking into account the internal structure of words and the out-of-vocabulary cases for unseen words. Therefore, one important improvement we have added in our study is that, instead of using a distinct vector representation for the DNA word, we took into account the internal structure of each word. Each DNA word is represented as a bag of character n-gram. By learning word embedding in this way, we can use the sub-word information that has proven to be effective in the field of natural language processing. The idea of taking into account the internal structure of DNA sequences is described in Fig. 2. We represent the DNA word (‘ATGAC’) by changing the n-gram values from 1 to 3 and the bags of nucleobases are different between the three types. As seen in Fig. 2, we are able to generate the sub-word information of every word in the dataset.

2.3. FastText implementation

We employed FastText tool [29] to generate the word embedding. FastText supports the training of continuous bag of words (CBOW) or skip-gram models using negative sampling, softmax or hierarchical softmax loss functions. The difference manifests between FastText and

Word2Vector model is as follows:

1. Generate better word embeddings for rare words (even if words are rare, their character n-grams are still shared with other words - hence the embeddings can still be valid).
2. Out of vocabulary words - they can construct the vector for a word from its character n-grams even if the word does not appear in training corpus. Both Word2vec and Glove cannot.
3. From a practical usage standpoint, the choice of hyper-parameters for generating FastText embeddings becomes key.
4. The usage of character embeddings (individual characters as opposed to n-grams) for downstream tasks have recently shown to be able to boost the performance of those tasks compared to using word embeddings like Word2Vec or Glove.

Each word is represented as a bag of character n-grams in addition to the word itself. This helps preserve the meaning of shorter words that may show up as n-grams of other words. Inherently, this also allows the capturing of meanings for suffixes/prefixes. Therefore, with the idea of using FastText in this study, we are able to achieve an outstanding performance for DNA sequence representations and classification, especially in the case of rare words by making use of character level information [29].

2.4. Feature extraction by FastText model

Supervised learning classification requires the input data to have the same number of features. As our DNA sequences are of different lengths, the number of biological words therefore also varies with the word embedded in DNA sequences. To address this, we set the dimension of the embedding vector to 100. This means that each DNA sequence is represented as 100 real numeric values and we can enter any machine learning classifier without preprocessing. Our rationale behind this idea is to highlight that the biological words can be analogous to the motifs in DNA sequences. Enhancers with similar motifs tend to have a higher degree of similarity. By taking the motif information into account via the word embedding, we have more informative features for better prediction.
There are two algorithms to create word vectors, i.e., CBOW and skip-gram. Given a set of sentences (also known as corpus), the model loops on the words of each sentence and either tries to use the current word to predict its neighbors (its context), in which the method is called “skip-gram” or it uses each of these contexts to predict the current word, in which the method is called “Continuous Bag Of Words” (CBOW).

The limit to the number of words in each context is determined by a parameter called “window size”. Our preliminary study uses both models, skip-gram, and CBOW to generate word embedding and SVM as a classifier. The results revealed that the skip-gram model generated a slightly higher performance than the CBOW. Consequently, we used the skip-gram model for all further experiments.

Here we provided a summary of the skip-gram model for completeness. As originally developed by Ref. [70], given a sequence of nucleobases \( n_1, n_2 \ldots n_T \), the skip-gram model aims to maximize the average log-likelihood:

\[
\frac{1}{T} \sum_{i=1}^{T} \log p(n_{i+1} | n_i)
\]

(1)

where \( T \) is the total number of nucleobases in the whole enhancer dataset, \( c \) is the context window size (i.e. the number of nucleobases on the left and right side of the target nucleobase), and \( p(n_t+j|nt) \) is defined as:

\[
p(n_t|n_t) = \frac{\exp(v_{nt}, v_t)}{\sum_k \exp(v_{nt}, v_k)}
\]

(2)

where \( v_t \) and \( v_{nt} \) are two space vector representations of the nucleobase \( n \). The subscripts \( 0 \) and \( c \) correspond to the output (context nucleobases) nucleobase and input (target) nucleobase, respectively. \( N \) is the total number of single nucleobases in the DNA vocabulary. In the typical NLP text corpus with a large vocabulary, the calculation of the log likelihood gradient becomes impractical. An approximation of the log likelihood is obtained by replacing every \( \log p(n_0|n) \) with:

\[
\log \sigma(v_{nt}, v_t) + \sum_{i=1}^{k} E_{nt-n_t}[\log \sigma(v_{nt}, v_t)]
\]

(3)

where

\[
\sigma(x) = \frac{1}{1 + \exp(-x)}
\]

(4)

and \( k \) are negative samples. This was motivated by the idea that a good model should be able to differentiate real data from negative data.

By treating DNA sequences as standard language sentences in a text corpus, NLP algorithms can be easily applied. More specifically, DNA sequences are treated as individual sentences, and nucleobases are treated as words. In this study, the skip-gram model is used with 100-dimensional vector space integration, a context window of size 8 and 6 for two-layer predictions, respectively.

2.5. Support vector machine

Support vector machine (SVM) is a well-known supervised learning tool to solve classification and regression problems with high performance. Given a number of training examples, each was marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one or the other category, making it a non-probabilistic binary linear classifier. SVM has been increasingly applied to a variety of supervised learning classifications in bioinformatics [71–73]. According to a binary classification, the SVM maps input samples into a higher dimensional space using the kernel function and then finds a hyperplane to distinguish between the two classes with maximum margin and minimum error. In this study, we implemented the SVM algorithm using the scikit-learn package in python language. We selected the radial basis function (RBF) as the learning kernel function due to many evidential improvements with this kernel [74,75]. We also performed a grid search to estimate the accuracy of each parameter combination to find the optimal cost and gamma in SVM (log2c was ranged from \(-5\) to \(15\) (step = \(2\)), log2g was ranged from \(3\) to \(-15\) (step = \(-2\))).

2.6. Assessment of predictive ability

Although the jackknife test is an approximately unbiased estimator of the generalization performance, it has two major drawbacks, e.g. a high variance (since all the datasets used for the estimate are very similar to each other), and it is expensive to calculate (it requires \( n \) estimates, where \( n \) is the number of observations in the dataset). It is also proven that the 5 or 10 fold cross-validation is a good compromise between impartiality and computational requirements. Not only that, other forms of resources to learn more about the cross-validation process are being highlighted [76]. Likewise, we first trained the model by applying 5-fold cross-validation technique to the whole training dataset. Because 5-fold cross-validation will yield different results every time, we implemented ten times of 5-fold cross-validation to generate more reliable results. The cross-validation result was being calculated through the averaging of all the ten 5-fold cross-validation results. Based on the 5-fold cross-validation results, hyper-parameter optimization process was employed to find the best model for each dataset. Furthermore, to prevent any systematic bias in the cross-validation set, the independent dataset was re-used to evaluate the performance accuracy.

In a bid to evaluate the performance of the methods employed, we adopted Chou’s criterion which has been used in many computational biology studies [21,77–80]. Although the traditional metrics copied from math books were often used in literature to measure the prediction quality of a prediction method, they are no longer an optimum method because of the lack of intuitiveness and difficulty in understanding for most biologists. Particularly the MCC (the Matthews correlation coefficient), which is a very important metrics used for reflecting the stability of a prediction method. Fortunately, based on the Chou’s symbols introduced for studying protein signal peptides [81–83], a set of four intuitive metrics were derived [84,85], as given in Eqs. (5)–(8). Ever since then, the new set of intuitive metrics has been with and admired by a series of recent publications (see, e.g. Refs. [37,66,84,86,87]). However, it is instructive to point out that the metrics (as defined in Eqs. (5)–(8) and their original non-intuitive forms) are only valid for single label systems. Multi-label systems (where a sample may simultaneously belong to several different classes), whose existence have become more frequent in system biology [34,88–91], system medicine [92,93] and biomedicine [94], a completely different set of metrics as defined in Ref. [95] is absolutely needed. Some standard metrics were used, such as sensitivity, specificity, accuracy, and MCC using below given formulae (TP, FP, TN, FN are true positive, true negative, and false negative, respectively):

\[
\text{Sensitivity} = 1 - \frac{N^+}{N^+} , \quad 0 \leq \text{Sen} \leq 1
\]

(5)

\[
\text{Specificity} = 1 - \frac{N^-}{N^-} , \quad 0 \leq \text{Spec} \leq 1
\]

(6)

\[
\text{Accuracy} = 1 - \frac{N^+ + N^-}{N^+ + N^-} , \quad 0 \leq \text{Acc} \leq 1
\]

(7)

\[
\text{MCC} = \frac{1 - \frac{N^+/N^-}{N^+ + N^-}}{\sqrt{(1 + \frac{N^+/N^-}{N^+ + N^-}) (1 + \frac{N^-/N^+}{N^- + N^+})}} , \quad -1 \leq \text{MCC} \leq 1
\]

(8)

The relations between these symbols and the symbols in Eqs. (5)–(8) are given by:
We also used the Receiver Operating Characteristic (ROC) curves to further illustrate our model in various experiments. In addition, the area under the ROC curve (AUC) metric is a scalar value that represents the overall performance of the model [96]. The AUC score is always bounded between zero and one, and there is no realistic classification with an AUC less than 0.5. The AUC metric is therefore used to compare the efficiency of different models.

3. Results and discussions

3.1. Classifying enhancers with different n-gram levels

We evaluated the performance of the SVM algorithm on the dataset of which, each enhancer sequence was split to biological words of equal length from 1 to 10. We used ROC Curve and AUC as the metrics to evaluate the overall performance of each experiment in both 5-fold cross-validation and independent test. The result of the first layer (enhancer identification) is displayed in Fig. 3. From Fig. 3, we found that the performance is proportional to the number of n-gram values. If we solely used the low levels of n-gram values (1, 2, or 3), the results would not reach its optimum performance. We realized that when n-gram values are greater than 6, the AUC is more consistent with almost no improvement. This means that the model only captures the information in a range of n-gram, increasing high level of n-gram does not help to in the increment of the results. As shown in Fig. 3, we chose n-gram = 8 with better AUC (≈ 0.91) to perform further experiments. Similar, Fig. 4 shows the performance results in the second layer and we chose n-gram = 6 to build our model.

3.2. Comparison between support vector machine and baseline models

FastText provides a multinomial logistic regression, where the sentence/document vector corresponds to the features. It has been integrated and used in many text classification problems [30], therefore we would like to compare it with their baseline classifier. We used the same dataset with the optimal n-gram levels (8-g for the first layer and 6-g for the second layer). The next baseline is gapped k-mer SVM (gkmSVM), which is a sequence-based method for predicting regulatory DNA elements [97,98]. We also searched for the optimal parameters for both classifiers. For FastText, the optimal parameters are: learning rate of 0.1, epoch of 50, context window of 5, and softmax loss function). The purpose of this analysis phase is to find out which classifier would attain the best performance with the embedding features. The performance results of all the experiments are shown in Table 1. The problem raised here is how to sustain the better performance of our algorithm, as compared to other baselines when it undergoes many cross-validation tests. To answer this question, we performed a paired t-test to determine whether our SVM is significantly better (+) or worse (−), or even have no statistical difference compared to the other baseline methods. The null hypothesis assumes that the true mean difference between the paired metrics is zero and the statistical significance is determined by p-value = 0.05 (confidence level of 95%). As shown in Table 1, it is clear that the SVM classifier outperforms FastText and gkmSVM on the same proposed dataset. Therefore, creating different algorithms based on their vectors is better than using a baseline model. Also, our performance results are also better than gkmSVM at the same level comparison.

3.3. Independent test

The most important concern out of all machine learning problems is the problem of overfitting, which means our classifier can only perform well in training set but worse in another unseen dataset. Therefore, we used an independent test to ensure that our model also performs well in a blind dataset. The independent dataset as described in the previous section was collected from Refs. [19,21]. It contained 200 enhancers (100 weak and 100 strong enhancers) and 200 non-enhancers. None of these samples have a certain occurrence in the training set. As shown in Table 2, our independent test results are also consistent with cross-validation results. In the first layer, our model reached an accuracy of 79%, sensitivity of 82%, specificity of 76%, and MCC of 0.58. Subsequently, the second layer's accuracy, sensitivity, specificity, and MCC reached 63.5%, 74%, 53%, and 0.28, respectively. Although there are a
few overfittings in our model, the differences are not minimal and it still shows that our model performs well in this type of dataset.

3.4. Comparison between proposed method and the existing predictors

So far, our best model is the combination of the support vector machine classifier and the n-gram level of 8 and 6 for two-layer classifications, respectively. In this section, we aim to compare the effectiveness of our proposed features with other baseline methods as well as research groups studying on enhancer classification problem, e.g. EnhancerPred [20], iEnhancer-2L [19], and iEnhancer-EL [21]. We compared both cross-validation and independent test performance. The results are shown in Table 2, with the highest values for each metrics of each class bolded. It is clear that on an average, our method outperforms other's models in almost measurement metrics. Therefore, we are able to present an innovative method for extracting features in DNA sequences in which outperforms other feature sets.

Although our word embedding features are able to produce encouraging results, further improvements to the performance results can still be made. One of the most important improvements that can be done is to integrate long-range information, which is a feature that plays an important role in DNA element identification. It has been successfully used in a number of recent bioinformatics problems on DNA sequences, such as identifying origin of replication in Saccharomyces cerevisiae [99] and predicting bacterial transcriptional terminators [100]. Therefore, its future works could consider combining long-range correlation information with word embedding feature to improve accuracy. Another solution is to take advantage of the development of deep learning. As deep learning has been applied in the other bioinformatics problems with significant results [101–104], enhancer identification is also a potential application. For instance, future works could also create a system by replacing our traditional SVM with a deep neural network. This is a promising area of research because of the ever increasing implementation of deep neural network structures to other forms of research.

3.5. Web server for identifying enhancers and their strength

As pointed out in Ref. [105] and demonstrated in a series of recent publications (see, e.g. Refs. [34,84,86,88,92,106–111]), user-friendly and publicly accessible web-servers represent the future direction for developing more practical and useful prediction methods as well as computational tools. Actually, many practical and useful web-servers have significantly increased the impacts of bioinformatics on medical science [47], driving medicinal chemistry into an unprecedented revolution [59], thus, we developed a web server named iEnhancer-SStep.

### Table 1

Comparison between SVM and FastText classifier.

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Sn</th>
<th>Sp</th>
<th>Acc</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st layer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FastText</td>
<td>72.3</td>
<td>81.4</td>
<td>76.9</td>
<td>0.54</td>
</tr>
<tr>
<td>gkmSVM</td>
<td>73.9</td>
<td>79.6</td>
<td>76.8</td>
<td>0.54</td>
</tr>
<tr>
<td>SVM</td>
<td>81.1 (+)</td>
<td>83.5 (+)</td>
<td>82.3 (+)</td>
<td>0.65 (+)</td>
</tr>
<tr>
<td>2nd layer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FastText</td>
<td>72.3</td>
<td>58.8</td>
<td>65.5</td>
<td>0.31</td>
</tr>
<tr>
<td>gkmSVM</td>
<td>67.8</td>
<td>50.0</td>
<td>59.8</td>
<td>0.18</td>
</tr>
<tr>
<td>SVM</td>
<td>75.3 (+)</td>
<td>60.8 (+)</td>
<td>68.1 (+)</td>
<td>0.37 (+)</td>
</tr>
</tbody>
</table>

(Statistical paired t-test results between SVM and other two methods: (+) for significantly better, (−) for significantly worse than others).

### Table 2

Comparison with previous predictors.

<table>
<thead>
<tr>
<th>Predictors</th>
<th>Cross-validation</th>
<th>Independent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sn</td>
<td>Sp</td>
</tr>
<tr>
<td>1st layer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>iEnhancer-5Step</td>
<td>81.1</td>
<td>83.5</td>
</tr>
<tr>
<td>iEnhancer-EL</td>
<td>75.67</td>
<td>80.39</td>
</tr>
<tr>
<td>iEnhancer-2L</td>
<td>78.09</td>
<td>75.88</td>
</tr>
<tr>
<td>EnhancerPred</td>
<td>72.57</td>
<td>73.79</td>
</tr>
<tr>
<td>2nd layer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>iEnhancer-5Step</td>
<td>75.3</td>
<td>60.8</td>
</tr>
<tr>
<td>iEnhancer-EL</td>
<td>69</td>
<td>61.05</td>
</tr>
<tr>
<td>iEnhancer-2L</td>
<td>62.21</td>
<td>61.82</td>
</tr>
<tr>
<td>EnhancerPred</td>
<td>62.67</td>
<td>61.46</td>
</tr>
</tbody>
</table>

Fig. 4. Performance results on classifying the strength of enhancers with different n-gram levels. The results with 6-g and 10-g (AUC = 0.75) outperforms other n-gram levels.
in our study. It is a web server specially trained for classifying enhancers by using SVM algorithm and biological word embedding as its main feature. Based on this web server, biologists are able to discover new sequences belonging to the enhancers as well as their strengths. In addition, to maximize the convenience of most experimental scientists, we have provided a step-by-step guide in which the users can easily obtain their desired results without having to go through the mathematical details.

(1) Click the link at http://www.biologydeep.com/fastenc to open the web-server iEnhancer-S5Step.
(2) Scroll down to submission field or click the tab ‘Submit’. In order to avoid the errors, please submit the sequence in FASTA format (we also give you the FASTA file examples). The user can choose two options to submit: pasting the sequence into the text field or uploading the sequence file. The user can submit one single FASTA file or multiple FASTA files.
(3) Click on the Submit button to see the predicted results.
(4) In the result page, the results for the sequences which belong to enhancers as well as their strengths will be shown. In case the sequence was predicted as non-enhancer, our server will display their strength as “NA”.

4. Conclusion

Improving the classification of enhancers is an essential task of biological researchers. Many studies were conducted using a variety of feature extractions and neural networks in a bid to resolve this problem. Looking at the outstanding results of word embedding in natural language processing, applying it to DNA sequence prediction is crucial for biological researchers. In this study, we took on an innovative method to classify the enhancers, using word embedding with sub-word information and SVM. This is the first study that applies this method to DNA sequence classification. With this method, we can interpret the DNA sequences as biological words and improve predictive performance. We evaluated the performance using 5-fold cross-validation and independent testing dataset. On an average, our method showed a 5-fold cross-validation accuracy of 82.3% and 68.1% in the first and second layer prediction, respectively. We also performed two independent tests to evaluate the performance of these two layer classifications, and the results achieved the accuracy of 79% and 63.1%, respectively. Compared with the performance of other state-of-the-art predictors, this approach achieved an evident improvement in almost all of the measurement metrics. Throughout this study, we have taken on this approach that uses a powerful model for the classification of enhancers and yielding high accuracy. The findings of this study could open a platform for further research that can interpret the biological words in DNA sequences. Moreover, scientists can use our approach to solve various computational biology problems in the future.

Acknowledgments

This work has been supported by the Nanyang Technological University Start-Up Grant.

References


