Bioinformatics, 34(22), 2018, 3835–3842
doi: 10.1093/bioinformatics/bty458
Advance Access Publication Date: 7 June 2018
Original Paper

OXFORD

Sequence analysis

iEnhancer-EL: identifying enhancers and their strength with ensemble learning approach

Bin Liu[1,2,*], Kai Li[1], De-Shuang Huang[3,*] and Kuo-Chen Chou[2,4,*]

[1]School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, Guangdong 518055, China, [2]Gordon Life Science Institute, Belmont, MA 02478, USA, [3]Institute of Machine Learning and Systems Biology, School of Electronics and Information Engineering, Tongji University, Shanghai 201804, China and [4]Center for Informational Biology, University of Electronic Science and Technology of China, Chengdu 610054, China

*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on January 23, 2018; revised on May 16, 2018; editorial decision on June 1, 2018; accepted on June 6, 2018

Abstract

Motivation: Identification of enhancers and their strength is important because they play a critical role in controlling gene expression. Although some bioinformatics tools were developed, they are limited in discriminating enhancers from non-enhancers only. Recently, a two-layer predictor called 'iEnhancer-2L' was developed that can be used to predict the enhancer's strength as well. However, its prediction quality needs further improvement to enhance the practical application value.

Results: A new predictor called 'iEnhancer-EL' was proposed that contains two layer predictors: the first one (for identifying enhancers) is formed by fusing an array of six key individual classifiers, and the second one (for their strength) formed by fusing an array of ten key individual classifiers. All these key classifiers were selected from 171 elementary classifiers formed by SVM (Support Vector Machine) based on kmer, subsequence profile and PseKNC (Pseudo K-tuple Nucleotide Composition), respectively. Rigorous cross-validations have indicated that the proposed predictor is remarkably superior to the existing state-of-the-art one in this area.

Availability and implementation: A web server for the iEnhancer-EL has been established at http://bioinformatics.hitsz.edu.cn/iEnhancer-EL/, by which users can easily get their desired results without the need to go through the mathematical details.

Contact: bliu@hit.edu.cn or dshuang@tongji.edu.cn or kcchou@gordonlifescience.org

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Enhancers are noncoding DNA fragments but they play a key role in controlling gene expression for the production of RNA and proteins (Omar *et al.*, 2017). Enhancers can be located up to 20 kb away from a gene, or even in a different chromosome (Liu *et al.*, 2016a); while promoters (a kind of gene proximal elements) are located near the transcription start sites of genes. Such locational difference makes the identification of enhancers much more challenging than that of promoters.

In the earlier days, identification of enhancers was carried out purely by the experimental techniques, such as the pioneering works reported in Heintzman and Ren, (2009) and (Boyle *et al.* (2011). The former was to detect enhancers via their combination with TF (transcription factor) such as P300 (Heintzman *et al.*, 2007; Visel *et al.*, 2009), and hence it would miss or under-detect the targets concerned because not all enhancers are occupied by TFs, resulting in high false negative rate (Chen *et al.*, 2007). The latter was to identify enhancers via the DNase I hypersensitivity, and hence some

other DNA segments or non-enhancers might be incorrectly or over detected as enhancers (Liu *et al.*, 2016aLiu et al., 2018b), leading to high false positive rate (Chen *et al.*, 2007). Although the follow-up techniques of genome-wide mapping of histone modifications (Ernst *et al.*, 2011; Erwin *et al.*, 2014; Fernández and Miranda-Saavedra, 2012; Firpi *et al.*, 2010; Kleftogiannis *et al.*, 2015; Rajagopal *et al.*, 2013) can alleviate the aforementioned shortcomings in detecting the enhancers and promoters and improve the detection rate, they are expensive and time-consuming.

In order to fast identify enhancers in genomes, several computational prediction methods have been developed, including CSI-ANN (Firpi *et al.*, 2010), EnhancerFinder (Erwin *et al.*, 2014), RFECS (Rajagopal *et al.*, 2013), EnhancerDBN (Bu *et al.*, 2017) and BiRen (Yang *et al.*, 2017). These bioinformatics tools differ with each other in using different sample formulation and/or operational algorithm during the 2nd and/or 3rd steps of the 5-step rule (Chou, 2011). For instance: CSI-ANN (Firpi *et al.*, 2010) is featured by using 'efficient data transformation' to formulate the samples, and the algorithm of Artificial Neural Network (ANN); EnhancerFinder (Erwin *et al.*, 2014) is featured by incorporating the evolutionary conservation information into the sample formulation, and the combined multiple kernel learning algorithm; RFECS (Rajagopal *et al.*, 2013), featured by the random forest algorithm (Rajagopal *et al.*, 2013); EnhancerDBN (Bu *et al.*, 2017) is based on the deep belief network; BiRen (Yang *et al.*, 2017) improved the predictive performance by using deep learning techniques. Using these bioinformatics tools, users can easily obtain their desired data. However, enhancers are a large group of functional elements formed by many different subgroups (Shlyueva *et al.*, 2014), such as strong enhancers, weak enhancers, poised enhancers, inactive enhancers, etc. The iEnhancer-2L (Liu *et al.*, 2016a) is the first predictor ever developed that is able to identify both the enhancers and their strength based only on the sequence information alone, and hence has been increasingly used in the genomics analysis. The iEnhancer-2L (Liu *et al.*, 2016a) is featured by the Pseudo K-tuple nucleotide composition (PseKNC) (Chen *et al.*, 2014, 2015a). Later, this method was further improved by incorporating other sequence-based features, for examples, the EnhancerPred (Jia, 2016 #45), bi-profile Bayes (Shao *et al.*, 2009), pseudo-nucleotide composition (Chen *et al.*, 2014), EnhancerPred2.0 (He and Jia, 2017) and electron–ion interaction pseudopotentials of nucleotides (Nair and Sreenadhan, 2006).

However, the success rates of these predictors need to be further improved, particularly in discriminating the strong enhancers from the weak ones. This study was initiated in an attempt to deal with this problem.

According to the Chou's 5-step rules (Chou, 2011) that have been followed by a series of recent studies (see e.g. Cheng *et al.*, 2018a; Feng *et al.*, 2017; Liu *et al.*, 2017a,b,c, 2018b; Song *et al.*, 2018b; Xiao *et al.*, 2017; Xu *et al.*, 2017), to develop a really useful predictor for a biological system, one should make the following five steps logically very clear: (i) benchmark dataset construction or selection, (ii) sample formulation, (iii) operation engine or algorithm, (iv) cross-validation and (v) web-server.

Below, let us elaborate the five steps one by one.

2 Materials and methods

2.1 Benchmark dataset

For facilitating comparison, the benchmark dataset $\mathbb{S}$ used in this study was taken from (Liu *et al.*, 2016a) that can be formulated as

$$\begin{cases} \mathbb{S} = \mathbb{S}^+ \cup \mathbb{S}^- \\ \mathbb{S}^+ = \mathbb{S}^+_{\text{strong}} \cup \mathbb{S}^+_{\text{weak}} \end{cases} \tag{1}$$

where the subset $\mathbb{S}^+$ contains 1484 enhancer samples, $\mathbb{S}^-$ contains 1484 non-enhancer samples, $\mathbb{S}^+_{\text{strong}}$ contains 742 strong enhancer samples, $\mathbb{S}^-_{\text{weak}}$ contains 742 weak enhancer samples, and $\cup$ is the symbol for union in the set theory. For readers' convenience, the detailed sequences for the aforementioned samples are given in Supplementary Information S1.

2.2 Sample formulation

One of the prerequisites in developing an effective bioinformatics predictor is how to formulate a biological sequence with a discrete model or a vector, yet still considerably keep its sequence-order information or key pattern characteristic. This is because all the existing machine-learning algorithms can only handle vectors but not sequences, as elucidated in a comprehensive review (Chou, 2015). However, a vector defined in a discrete model may completely lose all the sequence-pattern information (Chou, 2001a). To avoid this, here the DNA sequence samples were converted into vectors via the BioSeq-Analysis tool (Liu, 2018) to incorporate the information of kmer (Liu *et al.*, 2016b), subsequence profile (Lodhi *et al.*, 2002; Luo *et al.*, 2016; Yasser *et al.*, 2008) and pseudo *k*-tuple nucleotide composition (PseKNC) (Chen *et al.*, 2014, 2015b), as detailed below.

2.2.1 Kmer

Kmer (Liu *et al.*, 2016b) is the simplest approach to represent the DNA sequences, in which the DNA sequences are represented as the occurrence frequencies of k neighbouring nucleic acids. According to the sequential model, a DNA sample with L nucleotides is generally expressed by

$$\mathbf{D} = \mathrm{N}_1\mathrm{N}_2 \cdots \mathrm{N}_i \cdots \mathrm{N}_L \tag{2}$$

where N_1 denotes the 1st nucleotide at the sequence position 1, N_2 the 2nd nucleotide at the position 2 and so forth. They can be any of the four nucleotides; i.e.

$$\mathrm{N}_i \in \{\mathrm{A}\ (\text{adenine})\quad \mathrm{C}\ (\text{cytosine})\quad \mathrm{G}\ (\text{guanine})\quad \mathrm{T}\ (\text{thymine}) \tag{3}$$

where $\in$ is a symbol in the set theory meaning 'member of'. If using kmer to represent the DNA sequence of Eq. 2, we have (Chen *et al.*, 2014; Liu *et al.*, 2015)

$$\mathbf{D} = \left[f_1^{\text{kmer}} \quad f_2^{\text{kmer}} \quad \cdots \quad f_i^{\text{kmer}} \quad \cdots \quad f_{4^k}^{\text{kmer}} \right]^{\mathbf{T}} \tag{4}$$

where $f_i^{\text{kmer}} \left(i = 1,\ 2,\ \cdots,\ 4^k\right)$ is the occurrence frequencies of k neighbouring nucleotides in the DNA sequence $\mathbf{D}$ and $\mathbf{T}$ is the transpose operator. For example, when $i = 3$, Eq. 4 will become a 3mer vector

$$\begin{aligned} \mathbf{D} &= [f(\mathrm{AAA}) \quad f(\mathrm{AAC}) \quad f(\mathrm{AAT}) \quad \cdots \quad f(\mathrm{TTT})]^{\mathbf{T}} \\ &= \left[f_1^{3\text{mer}} \quad f_2^{3\text{mer}} \quad f_3^{3\text{mer}} \quad \cdots \quad f_{64}^{3\text{mer}} \right]^{\mathbf{T}} \end{aligned} \tag{5}$$

There is one parameter (k) in the kmer approach.

2.2.2 Subsequence profile

The subsequence profile (Lodhi *et al.*, 2002; Luo *et al.*, 2016; Yasser *et al.*, 2008) allows non-continuous mismatching, which may improve the Kmer approach in dealing with the cases of residue mutation, deletion and replacement during the biological sequence

evolutionary process. Its detailed formulation has been clearly elaborated in Luo *et al.* (2016), and hence there in no need to repeat here.

The subsequence profile contains two parameters k and δ; the latter is used to reflect the mismatch's extent (Luo *et al.*, 2016).

2.2.3 Pseudo k-tuple nucleotide composition

According to the pseudo k-tuple nucleotide composition or PseKNC (Chen *et al.*, 2014), the DNA sequence of Eq. 2 can be formulated as

$$\mathbf{D} = \left[f_1^{\text{PseKNC}} \quad f_2^{\text{PseKNC}} \quad \cdots \quad f_{4^k}^{\text{PseKNC}} \quad f_{4^k+1}^{\text{PseKNC}} \quad \cdots \quad f_{4^k+\lambda}^{\text{PseKNC}} \right]^{\text{T}} \tag{6}$$

where each of the components as well as the parameters k and λ have been very clearly defined in an original paper (Chen *et al.*, 2014) and a comprehensive review (Chen *et al.*, 2015a) via a series of sophisticated equations, and there is no need to repeat here. The essence is: it is through PseKNC that we are able to incorporate into Eq. 6 both the short-range or local sequence order information (via kmer) and the long-range or global sequence pattern information [via the concept of pseudo components (Chou, 2001a) and the six physicochemical properties of the dinucleotide in DNA (Chen *et al.*, 2014) as given in Supplementary Information S2]. In this study, these properties were normalized following the method reported in Chen *et al.* (2014).

There are three parameters in PseKNC (Chen *et al.*, 2014): k, w (the weight factor) and λ [the number of sequence correlations considered (Chou, 2005)].

2.3 Operation engine

In this study we chose to use SVM (Support Vector Machine) to operate the prediction. SVM is a machine-learning algorithm that has been widely used in the realm of bioinformatics (see e.g. Chen *et al.*, 2013, 2016; Ehsan *et al.*, 2018; Khan *et al.*, 2017; Liu *et al.*, 2014; Meher *et al.*, 2017; Rahimi *et al.*, 2017; Tahir *et al.*, 2017). For a brief formulation of SVM and how it works, see the papers (Cai *et al.*, 2003; Chou and Cai, 2002) without the need to repeat here. For more details about SVM, see a monograph (Cristianini and Shawe-Taylor, 2000).

The LIBSVM package (Chang and Lin, 2011) with the radial basis function (RBF) kernel was used to implement the learning machine, in which there are two parameters C (for the regularization) and γ (for the kernel width), which will be given later via an optimization approach.

Accordingly, when using SVM on kmer, subsequence profile, or PseKNC, we have a total of $(2+1)=3$, $(2+2)=4$ or $(2+3)=5$ uncertain parameters, respectively. The values for the two SVM-related parameters C and γ are determined by the final optimization as will be given later.

For the kmer approach with

$$k = 1,\ 2,\ 3,\ 4,\ 5,\ 6 \tag{7}$$

we can form six elementary classifiers as denoted by

$$\mathbb{C}^0(i),\ (i = 1,\ 2,\ \cdots,\ 6) \tag{8}$$

For the subsequence profile approach with

$$\begin{cases} 1 \le k \le 3 & \text{with step gap } \triangle = 1 \\ 0.1 \le \delta \le 1 & \text{with step gap } \triangle = 0.2 \end{cases} \tag{9}$$

we can form 15 elementary classifiers denoted by

$$\mathbb{C}^0(i),\ (i = 7,\ 8,\ \cdots,\ 21) \tag{10}$$

For the PseKNC approach with

$$\begin{cases} 1 \le k \le 6 & \text{with step gap } \triangle = 1 \\ 0.1 \le w \le 1 & \text{with step gap } \triangle = 0.2 \\ 1 \le \lambda \le 17 & \text{with step gap } \triangle = 4 \end{cases} \tag{11}$$

we can form 150 elementary classifiers denoted by

$$\mathbb{C}^0(i),\ (i = 22,\ 23,\ \cdots,\ 171) \tag{12}$$

Therefore, we have a total of $(6+15+150)=171$ different elementary classifiers.

2.4 Ensemble learning

As demonstrated by a series of previous studies (Chou and Shen, 2006a; Jia *et al.*, 2015, 2016a; Liu *et al.*, 2016b, 2017a; Qiu *et al.*, 2017), the ensemble predictor formed by fusing an array of individual predictors via a voting system can yield much better prediction quality.

There are two fundamental issues for developing an ensemble-learning predictor: one is how to select the key individual classifiers from the elementary ones to reduce the noise, and the other is how to fuse the selected key classifiers into one final classifier. Inspired by the works (Lin *et al.*, 2014a; Liu *et al.*, 2016b, 2017a), the treatment for the issue has been elaborated in Lin *et al.* (2014a) and Liu *et al.* (2016b, 2017a). The essence is that using the 'affinity propagation clustering algorithm' (Frey and Dueck, 2007) to cluster the elementary classifiers into a set of groups (Fig. 1a) and how the key classifiers were selected from these groups (Fig. 1b). For those who are interested in the detailed process, see Supplementary Information S3.

By doing so, six key individual classifiers were obtained (Table 1) for the 1st-layer prediction to identify enhancers from non-enhancers, as formulated by

$$\mathbb{C}^1(i),\ (i = 1,\ 2,\ \cdots,\ 6) \tag{13}$$

For the 2nd-layer prediction, ten key individual classifiers (Table 2) were obtained, as formulated by

$$\mathbb{C}^2(i),\ (i = 1,\ 2,\ \cdots,\ 10) \tag{14}$$

By fusing the six key individual classifiers in Eq. 13 as done in (Chou and Shen, 2006b; Shen and Chou, 2009), we obtained the 1st-layer ensemble classifier as given by

$$\mathbb{C}^{\text{E1}} = \mathbb{C}^1(1) \forall \mathbb{C}^1(2) \forall \cdots \forall \mathbb{C}^1(6) = \forall_{i=1}^{6} \mathbb{C}^1(i) \tag{15}$$

Likewise, by fusing the ten key individual classifiers in Eq. 14, we obtained the 2nd-layer ensemble classifier given by

$$\mathbb{C}^{\text{E2}} = \mathbb{C}^2(1) \forall \mathbb{C}^2(2) \forall \cdots \forall \mathbb{C}^2(10) = \forall_{i=1}^{10} \mathbb{C}^2(i) \tag{16}$$

where the symbol $\forall$ in Eqs. 15 and 16 denotes the fusing operator. For more details about the process of fusing individual classifiers into an ensemble classifier, see a comprehensive review (Chou and Shen, 2007) where a clear description with a set of elegant equations are given and hence there is no need to repeat here. Meanwhile, the genetic algorithm (Mitchell, 1998) was used to optimize the weight factors on the benchmark datasets by setting the number of

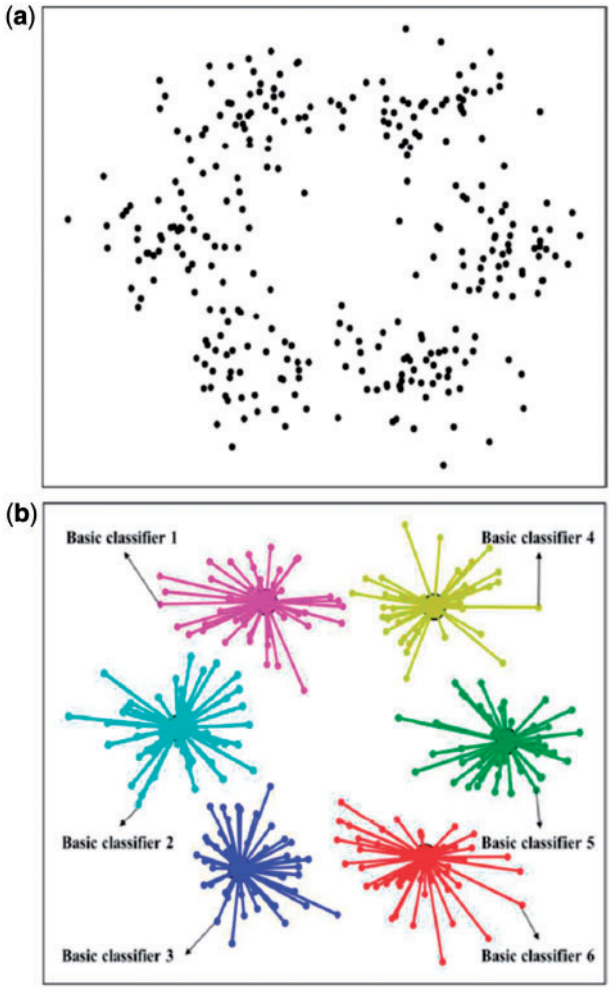

Fig. 1. An illustration to show (**a**) how the elementary classifiers were clustered into a set of groups, and (**b**) how to select the key classifiers from these groups

Table 1. List of the six key individual classifiers selected from the 171 elementary classifiers in Eqs. 8, 10 and 12 by using the affinity propagation clustering algorithm (Frey and Dueck, 2007) as done in (Liu *et al.*, 2016a) for the 1st-layer prediction

Key individual classifier	Feature vector	Dimension
$\mathbb{C}^1(1)$	PseKNC[a]	77
$\mathbb{C}^1(2)$	PseKNC[b]	81
$\mathbb{C}^1(3)$	PseKNC[c]	4113
$\mathbb{C}^1(4)$	Subsequence profile[d]	64
$\mathbb{C}^1(5)$	Kmer[e]	64
$\mathbb{C}^1(6)$	Kmer[f]	4096

[a]The parameters used: $k=3,\ \lambda=13, w=0.1,\ \mathrm{C}=2^6,\ \gamma=2^4$.
[b]The parameters used: $k=3,\ \lambda=17, w=0.1, \mathrm{C}=2^{10},\ \gamma=2^4$.
[c]The parameters used: $k=6,\ \lambda=17, w=0.1, \mathrm{C}=2^4,\ \gamma=2^5$.
[d]The parameters used: $k=3,\ \delta=0.5,\ \mathrm{C}=2^{-4},\ \gamma=2^{-9}$.
[e]The parameters used: $k=3, \mathrm{C}=2^4,\ \gamma=2^3$.
[f]The parameters used: $k=6, \mathrm{C}=2^1, \gamma=2^5$.

Table 2. List of the ten key individual classifiers selected from the 171 elementary classifiers in Eqs. 8, 10 and 12 by using the affinity propagation clustering algorithm (Frey and Dueck, 2007) as done in (Liu *et al.*, 2016a) for the 2nd-layer prediction

Key individual classifier	Feature vector	Dimension
$\mathbb{C}^2(1)$	PseKNC[a]	9
$\mathbb{C}^2(2)$	PseKNC[b]	9
$\mathbb{C}^2(3)$	PseKNC[c]	9
$\mathbb{C}^2(4)$	PseKNC[d]	13
$\mathbb{C}^2(5)$	PseKNC[e]	29
$\mathbb{C}^2(6)$	PseKNC[f]	77
$\mathbb{C}^2(7)$	PseKNC[g]	81
$\mathbb{C}^2(8)$	PseKNC[h]	265
$\mathbb{C}^2(9)$	Kmer[i]	64
$\mathbb{C}^2(10)$	Kmer[j]	4096

[a]The parameters used: $k=1,\ \lambda=5, w=0.1, \mathrm{C}=2^5,\ \gamma=2^2$.
[b]The parameters used: $k=1,\ \lambda=5, w=0.7,\ \mathrm{C}=2^3,\ \gamma=2^5$.
[c]The parameters used: $k=1,\ \lambda=5, w=0.9, \mathrm{C}=2^4,\ \gamma=2^5$.
[d]The parameters used: $k=1,\ \lambda=9, w=0.9,\ \mathrm{C}=2^3,\ \gamma=2^4$.
[e]The parameters used: $k=2,\ \lambda=13, w=0.1, \mathrm{C}=2^5,\ \gamma=2^5$.
[f]The parameters used: $k=3,\ \lambda=13, w=0.3, \mathrm{C}=2^4,\ \gamma=2^5$.
[g]The parameters used: $k=3,\ \lambda=17, w=0.7, \mathrm{C}=2^5,\ \gamma=2^5$.
[h]The parameters used: $k=5,\ \lambda=9, w=0.7, \mathrm{C}=2^4,\ \gamma=2^5$.
[i]The parameters used: $k=3,\ \mathrm{C}=2^3,\ \gamma=2^2$.
[j]The parameters used: $k=6, \mathrm{C}=2^1, \gamma=2^3$.

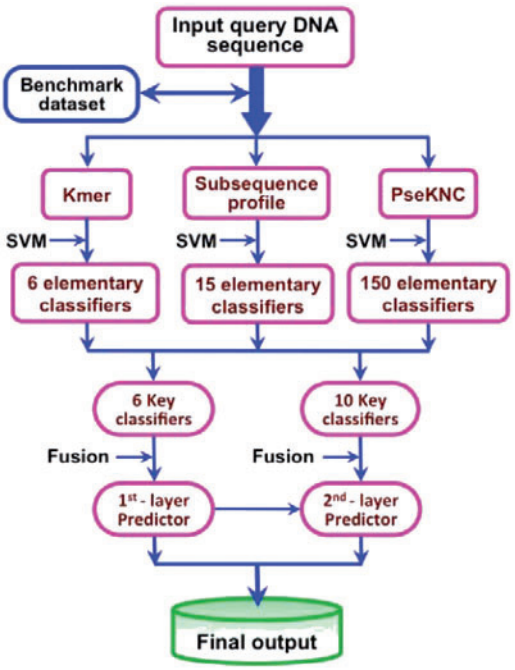

Fig. 2. A flowchart to illustrate how iEnhancer-EL is working

population size and evolutional generations as 200 and 2000 respectively for both the 1st and 2nd layers.

The proposed predictor for identifying enhancers and their strength is called iEnhancer-EL, where 'i' stands for 'identify' and 'EL' for 'ensemble learning'. In Figure 2 is a flowchart to illustrate how the predictor is working.

2.5 Cross-validation

To objectively evaluate the performance of a new predictor, we need to consider the following two issues: (i) what metrics should be used to reflect its performance in a quantitative way? (ii) what method should be adopted to derive the metrics?

In literature, the following four metrics are usually adopted to evaluate a predictor's quality (Chen *et al.*, 2007): (i) overall accuracy (Acc); (ii) stability (MCC); (iii) sensitivity (Sn); and (iv) specificity (Sp). But their formulations directly taken from math books are not intuitive and hence difficult to be understood by most biological scientists. However, by means of the symbols introduced by Chou in studying signal peptides (Chou, 2001b), the four metrics can be converted to a set of intuitive ones (Chen *et al.*, 2013; Xu *et al.*, 2013a) as given below:

$$\begin{cases} \mathrm{Sn} = 1 - \dfrac{N_-^+}{N^+} & 0 \le \mathrm{Sn} \le 1 \\ \mathrm{Sp} = 1 - \dfrac{N_+^-}{N^-} & 0 \le \mathrm{Sp} \le 1 \\ \mathrm{Acc} = 1 - \dfrac{N_-^+ + N_+^-}{N^+ + N^-} & 0 \le \mathrm{Acc} \le 1 \\ \mathrm{MCC} = \dfrac{1 - \left(\dfrac{N_-^+}{N^+} + \dfrac{N_+^-}{N^-}\right)}{\sqrt{\left(1 + \dfrac{N_+^- - N_-^+}{N^+}\right)\left(1 + \dfrac{N_-^+ - N_+^-}{N^-}\right)}} & -1 \le \mathrm{MCC} \le 1 \end{cases} \tag{17}$$

where N^+ represents the total number of positive samples investigated, while N_-^+ is the number of positive samples incorrectly predicted to be of negative one; N^- the total number of negative samples investigated, while N_+^- the number of the negative samples incorrectly predicted to be of positive one.

Based on the definition of Eq. 17, the meanings of Sn, Sp, Acc and MCC have become much more intuitive and easier to understand, as discussed and used in a series of recent studies in various biological areas (see e.g. Chen *et al.*, 2018a; Ehsan *et al.*, 2018; Feng *et al.*, 2017, 2018; Khan *et al.*, 2018; Liu *et al.*, 2017a,b,c, 2018a,b; Song *et al.*, 2018c; Xu *et al.*, 2014, 2017; Yang *et al.*, 2018). In addition, the Area Under ROC Curve (AUC) (Fawcett, 2006) was also used to measure quality of the predictor.

With a set of quantitative metrics clearly defined, the next is how to test their values. As is well known, the independent dataset test, subsampling (or K-fold cross-validation) test and jackknife test are the three cross-validation methods widely used for testing a prediction method (Chou and Zhang, 1995). To reduce the computational cost, in this study we adopted the 5-fold cross-validation (namely K = 5) to optimize the parameters in our method as done by many investigators with SVM as the prediction engine (see e.g. Khan *et al.*, 2017; Meher *et al.*, 2017; Rahimi *et al.*, 2017; Tahir *et al.*, 2017). The concrete process is as follows. The benchmark dataset was randomly divided into five subsets with an approximately equal number of samples. Each predictor runs five times with five different training and test sets. For each run, three sets were used to train the predictor, one set was used as the validation set to optimize the parameters, and the remaining one was used as the test set to give the predictive results. In this study, the jackknife test was also used to evaluate the performance of different methods.

3 Results and discussion

3.1 Comparison with the existing methods

Listed in Table 3 are the metrics rates (Eq. 17) achieved by iEnhancer-EL via the jackknife test on the benchmark dataset (cf. Supplementary Information S1). For facilitating comparison, listed there are also the corresponding rates obtained by iEnhancer-2L using exactly the same cross-validation method and same benchmark dataset.

Table 3. A comparison of the proposed predictor with the state-of-the-art predictor in identifying enhancers (the 1st-layer) and their strength (the 2nd-layer) via the jackknife test on the same benchmark dataset (Supplementary Information S1)

	Method	Acc(%)	MCC	Sn(%)	Sp(%)	AUC(%)
First layer	iEnhancer-EL[a]	78.03	0.5613	75.67	80.39	85.47
	iEnhancer-2L[b]	76.89	0.5400	78.09	75.88	85.00
	EnhancerPred[c]	73.18	0.4636	72.57	73.79	80.82
Second layer	iEnhancer-EL[a]	65.03	0.3149	69.00	61.05	69.57
	iEnhancer-2L[b]	61.93	0.2400	62.21	61.82	66.00
	EnhancerPred[c]	62.06	0.2413	62.67	61.46	66.01

[a]The predictor proposed in this paper.
[b]The predictor reported in Liu *et al.* (2016a).
[c]The predictor reported in Jia and He (2016).

From Table 3 we can see the following. (i) For the 1st-layer prediction, namely in discriminating enhancers from non-enhancers, except for Sn, the success rates achieved by the proposed predictor for the other metrics are all higher than those by the existing state-of-the-art predictors. (ii) For the 2nd-layer prediction, namely in identifying the strength of enhancers, except for Sp, all the other three metrics rates as well as the AUC value obtained by the proposed predictor are higher than those by the existing state-of-the art predictors. It is instructive to point out that, of the four metrics in Eq. 17, the most important are the Acc and MCC. The former is used to measure a predictor's overall accuracy, and the latter for its stability. Under such a circumstance, the iEnhancer-EL outperformed both iEnhancer-2L and EnhancerPred according to the Acc and MCC metrics.

3.2 Independent dataset test

An independent dataset was used to further evaluate the performance of various methods, which was constructed based on the same protocol as the one used in constructing the benchmark dataset. The independent dataset contains 100 strong enhancers, 100 weak enhancers and 200 non-enhancers (Supplementary Information S4). None of the samples in the independent dataset occurs in the training dataset. The CD-HIT software (Li and Godzik, 2006) was used to remove those samples in the independent dataset that have more than 80% sequence identity to any other in a same subset. The results obtained by the proposed predictor by the independent dataset test are given in Table 4, where for facilitating comparison, the corresponding results by other two methods were also listed. It can be clearly seen from the table that the iEnhancer-EL predictor is superior to its counterparts in nearly all the four metrics. Although the new predictor is slightly lower than iEnhancer-2L in Sp by 2.5%, its Sn rate is 4.5% higher than that of the iEnhancer-2L.

Note that, of the four metrics in Eq. 17, the most important are the Acc and MCC: the former reflects the overall accuracy of a predictor; while the latte, its stability in practical applications. The metrics Sn and Sp are used to measure a predictor from two different angles. When, and only when, both Sn and Sp of the predictor A are higher than those of the predictor B, can we say A is better than B. In other words, Sn and Sp are actually constrained with each other (Chou, 1993). Therefore, it is meaningless to use only one of the two for comparing the quality of two predictors. A meaningful

Table 4. A comparison of the proposed predictor with the state-of-the-art predictors in identifying enhancers (the 1st-layer) and their strength (the 2nd-layer) on the independent dataset (Supplementary Information S4)

	Method	Acc(%)	MCC	Sn(%)	Sp(%)	AUC(%)
First layer	iEnhancer-EL[a]	74.75	0.4964	71.00	78.50	81.73
	iEnhancer-2L[b]	73.00	0.4604	71.00	75.00	80.62
	EnhancerPred[c]	74.00	0.4800	73.50	74.50	80.13
Second layer	iEnhancer-EL[a]	61.00	0.2222	54.00	68.00	68.01
	iEnhancer-2L[b]	60.50	0.2181	47.00	74.00	66.78
	EnhancerPred[c]	55.00	0.1021	45.00	65.00	57.90

[a]The predictor proposed in this paper.
[b]The predictor reported in Liu *et al.* (2016a).
[c]The predictor reported in Jia and He (2016).

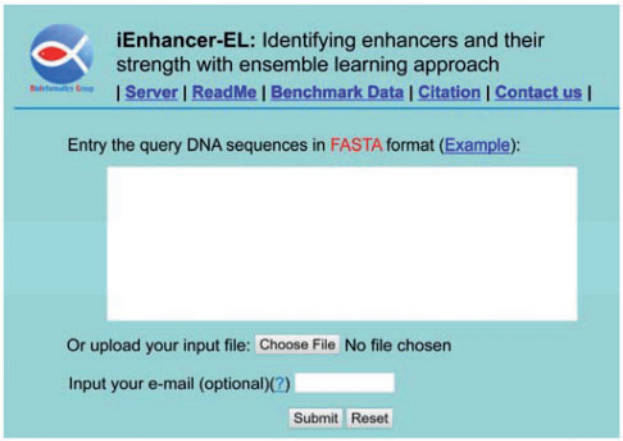

Fig. 3. A semi-screenshot to show the top page of iEnhancer-EL web server. Its web-site address is at http://bioinformatics.hitsz.edu.cn/iEnhancer-EL/

comparison in this regard should count the rates of both Sn and Sp, or even better the rate of their combination that is none but MCC, for which the proposed predictor achieved the highest rate as shown in Table 4.

3.3 Web-server and its user guide

As pointed out in (Chou and Shen, 2009) and supported by a series of follow-up publications (see e.g. Chen *et al.*, 2018b; Cheng *et al.*, 2017, 2018a,b; Jia *et al.*, 2015, 2016b; Lin *et al.*, 2014b; Liu *et al.*, 2018b; Song *et al.*, 2018a,b,c; Wang *et al.*, 2017, 2018; Xiao *et al.*, 2013; Xu *et al.*, 2013b), user-friendly and publicly accessible web-servers represent the future direction for developing practically more useful predictors. Actually, a new prediction method with the availability of a user-friendly web-server would significantly enhance its impacts (Chou, 2015), driving medicinal chemistry into an unprecedented revolution (Chou, 2017). In view of this, the web-server for iEnhancer-EL has been established. Furthermore, to maximize the convenience of most experimental scientists, the step-by-step instructions are given below.

Step 1. Open the web-server at http://bioinformatics.hitsz.edu.cn/iEnhancer-EL/ and you will see its top page as shown in Figure 3. Click on the Read Me button to see a brief introduction about the server.

Step 2. You can either type or copy/paste the query DNA sequence into the input box at the center of Figure 3, or directly upload your input data by the Browse button. The input sequence should be in the FASTA format. Not familiar with it? Click the Example button right above the input box.

Step 3. Click on the Submit button to see the predicted result. For example, if using the example sequence to run the web server, you will see the following outcome: (i) the first query sequence contains nine strong enhancers: sub-sequences 1-200, 2-201, 3-202, 4-203, 5-204, 6-205, 7-206, 8-207 and 9-208; (ii) the second query sequence contains one strong enhancer at sub-sequence 1-200; (iii) both the third and fourth query sequences contain one weak enhancer at sub-sequence 1-200; (iv) the fifth and sixth query sequences contain no enhancer. All these predicted results are fully consistent with experimental observations.

Step 4. You can download the predicted results into a file by clicking the Download button on the results page.

Acknowledgement

The authors are very much indebted to the four anonymous reviewers, whose constructive comments are very helpful for strengthening the presentation of this article.

Funding

This work was supported by the National Natural Science Foundation of China (No. 61672184, 61732012, 61520106006), Guangdong Natural Science Funds for Distinguished Young Scholars (2016A030306008), Scientific Research Foundation in Shenzhen (Grant No. JCYJ20170307152201596), Guangdong Special Support Program of Technology Young talents (2016TQ03X618), Fok Ying-Tung Education Foundation for Young Teachers in the Higher Education Institutions of China (161063) and Shenzhen Overseas High Level Talents Innovation Foundation (Grant No. KQJSCX20170327161949608).

Conflict of Interest: none declared.

References

Boyle,A.P. *et al.* (2011) High-resolution genome-wide in vivo footprinting of diverse transcription factors in human cells. *Genome Res.*, **21**, 456–464.
Bu,H. *et al.* (2017) A new method for enhancer prediction based on deep belief network. *BMC Bioinformatics*, **18**, 418.
Cai,Y.D. *et al.* (2003) Support vector machines for predicting membrane protein types by using functional domain composition. *Biophys. J.*, **84**, 3257–3263.
Chang,C.C. and Lin,C.J. (2011) LIBSVM: a Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.*, **2**, 1–27.
Chen,J. *et al.* (2007) Prediction of linear B-cell epitopes using amino acid pair antigenicity scale. *Amino Acids*, **33**, 423–428.
Chen,J. *et al.* (2016) dRHP-PseRA: detecting remote homology proteins using profile-based pseudo protein sequence and rank aggregation. *Sci. Rep.*, **6**, 32333.
Chen,W. *et al.* (2013) iRSpot-PseDNC: identify recombination spots with pseudo dinucleotide composition. *Nucleic Acids Res.*, **41**, e68.
Chen,W. *et al.* (2014) PseKNC: a flexible web-server for generating pseudo K-tuple nucleotide composition. *Anal. Biochem.*, **456**, 53–60.
Chen,W. *et al.* (2015a) Pseudo nucleotide composition or PseKNC: an effective formulation for analyzing genomic sequences. *Mol. BioSyst.*, **11**, 2620–2634.
Chen,W. *et al.* (2015b) PseKNC-General: a cross-platform package for generating various modes of pseudo nucleotide compositions. *Bioinformatics*, **31**, 119–120.
Chen,W. *et al.* (2018a) iRNA-3typeA: identifying 3-types of modification at RNA's adenosine sites. *Mol. Therapy Nucleic Acid*, **11**, 468–474.
Chen,Z. *et al.* (2018b) iFeature: a python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics*, doi: 10.1093/bioinformatics/bty140/4924718.

Cheng,X. *et al.* (2018a) pLoc-mEuk: predict subcellular localization of multi-label eukaryotic proteins by extracting the key GO information into general PseAAC. *Genomics*, **110**, 50–58.

Cheng,X. *et al.* (2018b) pLoc-mHum: predict subcellular localization of multi-location human proteins via general PseAAC to winnow out the crucial GO information. *Bioinformatics*, **34**, 1448–1456.

Cheng,X. *et al.* (2017) pLoc-mAnimal: predict subcellular localization of animal proteins with both single and multiple sites. *Bioinformatics*, **33**, 3524–3531.

Chou,K.C. (1993) A vectorized sequence-coupling model for predicting HIV protease cleavage sites in proteins. *J. Biol. Chem.*, **268**, 16938–16948.

Chou,K.C. (2001a) Prediction of protein cellular attributes using pseudo amino acid composition. *Proteins Struct. Funct. Genet. (Erratum: ibid., 2001, Vol.44, 60)*, **43**, 246–255.

Chou,K.C. (2001b) Prediction of protein signal sequences and their cleavage sites. *Proteins Struct. Funct. Genet.*, **42**, 136–139.

Chou,K.C. (2005) Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. *Bioinformatics*, **21**, 10–19.

Chou,K.C. (2011) Some remarks on protein attribute prediction and pseudo amino acid composition (50th Anniversary Year Review). *J. Theor. Biol.*, **273**, 236–247.

Chou,K.C. (2015) Impacts of bioinformatics to medicinal chemistry. *Med. Chem.*, **11**, 218–234.

Chou,K.C. (2017) An unprecedented revolution in medicinal chemistry driven by the progress of biological science. *Curr. Top. Med. Chem.*, **17**, 2337–2358.

Chou,K.C. and Cai,Y.D. (2002) Using functional domain composition and support vector machines for prediction of protein subcellular location. *J. Biol. Chem.*, **277**, 45765–45769.

Chou,K.C. and Shen,H.B. (2006a) Hum-PLoc: a novel ensemble classifier for predicting human protein subcellular localization. *Biochem. Biophys. Res. Commun. (BBRC)*, **347**, 150–157.

Chou,K.C. and Shen,H.B. (2006b) Predicting protein subcellular location by fusing multiple classifiers. *J. Cell. Biochem.*, **99**, 517–527.

Chou,K.C. and Shen,H.B. (2007) Review: recent progresses in protein subcellular location prediction. *Anal. Biochem.*, **370**, 1–16.

Chou,K.C. and Shen,H.B. (2009) Recent advances in developing web-servers for predicting protein attributes. *Nat. Sci.*, **01**, 63–92.

Chou,K.C. and Zhang,C.T. (1995) Review: prediction of protein structural classes. *Crit. Rev. Biochem. Mol. Biol.*, **30**, 275–349.

Cristianini,N. and Shawe-Taylor,J. (2000) *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Chapter 3*. Cambridge University Press, Cambridge, England.

Ehsan,A. *et al.* (2018) A novel modeling in mathematical biology for classification of signal peptides. *Sci. Rep.*, **8**, 1039.

Ernst,J. *et al.* (2011) Mapping and analysis of chromatin state dynamics in nine human cell types. *Nature*, **473**, 43–49.

Erwin,G.D. *et al.* (2014) Integrating diverse datasets improves developmental enhancer prediction. *PLoS Comput. Biol.*, **10**, e1003677.

Fawcett,J.A. (2006) An introduction to ROC analysis. *Pattern Recogn. Lett.*, **27**, 861–874.

Feng,P. *et al.* (2017) iRNA-PseColl: identifying the occurrence sites of different RNA modifications by incorporating collective effects of nucleotides into PseKNC. *Mol. Ther. Nucleic Acids*, **7**, 155–163.

Feng,P. *et al.* (2018) iDNA6mA-PseKNC: identifying DNA N6-methyladenosine sites by incorporating nucleotide physicochemical properties into PseKNC. *Genomics*, doi: 10.1016/j.ygeno.2018.01.005.

Fernández,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–e77.

Firpi,H.A. *et al.* (2010) Discover regulatory DNA elements using chromatin signatures and artificial neural network. *Bioinformatics*, **26**, 1579–1586.

Frey,B.J. and Dueck,D. (2007) Clustering by passing messages between data points. *Science*, **315**, 972–976.

He,W. and Jia,C. (2017) EnhancerPred2.0: predicting enhancers and their strength based on position-specific trinucleotide propensity and electron–ion interaction potential feature selection. *Mol. Biosyst.*, **13**, 767–774.

Heintzman,N.D. and Ren,B. (2009) Finding distal regulatory elements in the human genome. *Curr. Opin. Genet. Dev.*, **19**, 541–549.

Heintzman,N.D. *et al.* (2007) Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. *Nat. Genet.*, **39**, 311–318.

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.

Jia,J. *et al.* (2015) iPPI-Esml: an ensemble classifier for identifying the interactions of proteins by incorporating their physicochemical properties and wavelet transforms into PseAAC. *J. Theor. Biol.*, **377**, 47–56.

Jia,J. *et al.* (2016a) pSuc-Lys: predict lysine succinylation sites in proteins with PseAAC and ensemble random forest approach. *J. Theor. Biol.*, **394**, 223–230.

Jia,J. *et al.* (2016b) pSumo-CD: predicting sumoylation sites in proteins with covariance discriminant algorithm by incorporating sequence-coupled effects into general PseAAC. *Bioinformatics*, **32**, 3133–3141.

Khan,M. *et al.* (2017) Unb-DPC: identify mycobacterial membrane protein types by incorporating un-biased dipeptide composition into Chou's general PseAAC. *J. Theor. Biol.*, **415**, 13–19.

Khan,Y.D. *et al.* (2018) iPhosT-PseAAC: identify phosphothreonine sites by incorporating sequence statistical moments into PseAAC. *Anal. Biochem.*, **550**, 109–116.

Kleftogiannis,D. *et al.* (2015) DEEP: a general computational framework for predicting enhancers. *Nucleic Acids Res.*, **43**, e6–e6.

Li,W. and Godzik,A. (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, **22**, 1658–1659.

Lin,C. *et al.* (2014a) LibD3C: ensemble classifiers with a clustering and dynamic selection strategy. *Neurocomputing*, **123**, 424–435.

Lin,H. *et al.* (2014b) iPro54-PseKNC: a sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition. *Nucleic Acids Res.*, **42**, 12961–12972.

Liu,B. *et al.* (2014) Combining evolutionary information extracted from frequency profiles with sequence-based kernels for protein remote homology detection. *Bioinformatics*, **30**, 472–479.

Liu,B. (2018) BioSeq-Analysis: a platform for DNA, RNA, and protein sequence analysis based on machine learning approaches. *Brief. Bioinf.*, doi: 10.1093/bib/bbx165.

Liu,B. *et al.* (2015) repDNA: a Python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects. *Bioinformatics*, **31**, 1307–1309.

Liu,B. *et al.* (2016a) iEnhancer-2L: a two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition. *Bioinformatics*, **32**, 362–369.

Liu,B. *et al.* (2016b) iDHS-EL: identifying DNase I hypersensi-tivesites by fusing three different modes of pseudo nucleotide composition into an ensemble learning framework. *Bioinformatics*, **32**, 2411–2418.

Liu,B. *et al.* (2017a) iRSpot-EL: identify recombination spots with an ensemble learning approach. *Bioinformatics*, **33**, 35–41.

Liu,B. *et al.* (2017b) 2L-piRNA: a two-layer ensemble classifier for identifying piwi-interacting RNAs and their function. *Mol. Therapy Nucleic Acids*, **7**, 267–277.

Liu,L.M. *et al.* (2017c) iPGK-PseAAC: identify lysine phosphoglycerylation sites in proteins by incorporating four different tiers of amino acid pairwise coupling information into the general PseAAC. *Med. Chem.*, **13**, 552–559.

Liu,B. *et al.* (2018a) iRO-3wPseKNC: identify DNA replication origins by three-window-based PseKNC. *Bioinformatics*, doi: 10.1093/bioinformatics/bty312/4978052.

Liu,B. *et al.* (2018b) iPromoter-2L: a two-layer predictor for identifying promoters and their types by multi-window-based PseKNC. *Bioinformatics*, **34**, 33–40.

Lodhi,H. *et al.* (2002) Text classification using string kernels. *J. Mach. Learn. Res.*, **2**, 419–444.

Luo,L. *et al.* (2016) Accurate prediction of transposon-derived piRNAs by integrating various sequential and physicochemical features. *PLoS ONE*, **11**, e0153268.

Meher,P.K. *et al*. (2017) Predicting antimicrobial peptides with improved accuracy by incorporating the compositional, physico-chemical and structural features into Chou's general PseAAC. *Sci. Rep*., **7**, 42362.

Mitchell,M. (1998) *An Introduction to Genetic Algorithms*. MIT Press.

Nair,A.S. and Sreenadhan,S.P. (2006) A coding measure scheme employing electron–ion interaction pseudopotential (EIIP). *Bioinformation*, **1**, 197–202.

Omar,N. *et al*. (2017) Enhancer prediction in proboscis monkey genome: a comparative study. *J. Telecommun. Electron. Comput. Eng. (JTEC)*, **9**, 175–179.

Qiu,W.R. *et al*. (2017) iKcr-PseEns: identify lysine crotonylation sites in histone proteins with pseudo components and ensemble classifier. *Genomics*, doi: 10.1016/j.ygeno.2017.10.008.

Rahimi,M. *et al*. (2017) OOgenesis_Pred: a sequence-based method for predicting oogenesis proteins by six different modes of Chou's pseudo amino acid composition. *J. Theor. Biol*., **414**, 128–136.

Rajagopal,N. *et al*. (2013) RFECS: a random-forest based algorithm for enhancer identification from chromatin state. *PLoS Comput. Biol*., **9**, e1002968.

Shao,J. *et al*. (2009) Computational identification of protein methylation sites through bi-profile Bayes feature extraction. *PLoS One*, **4**, e4920.

Shen,H.B. and Chou,K.C. (2009) QuatIdent: a web server for identifying protein quaternary structural attribute by fusing functional domain and sequential evolution information. *J. Proteome Res*., **8**, 1577–1584.

Shlyueva,D. *et al*. (2014) Transcriptional enhancers: from properties to genome-wide predictions. *Nat. Rev. Genet*., **15**, 272–286.

Song,J. *et al*. (2018a) PROSPERous: high-throughput prediction of substrate cleavage sites for 90 proteases with improved accuracy. *Bioinformatics*, **34**, 684–687.

Song,J. *et al*. (2018b) PREvaIL, an integrative approach for inferring catalytic residues using sequence, structural and network features in a machine learning framework. *J. Theor. Biol*., **443**, 125–137.

Song,J. *et al*. (2018c) iProt-Sub: a comprehensive package for accurately mapping and predicting protease-specific substrates and cleavage sites. *Brief. Bioinf*., doi: 10.1093/bib/bby028.

Tahir,M. *et al*. (2017) Sequence based predictor for discrimination of enhancer and their types by applying general form of Chou's trinucleotide composition. *Comput. Methods Programs Biomed*., **146**, 69–75.

Visel,A. *et al*. (2009) ChIP-seq accurately predicts tissue-specific activity of enhancers. *Nature*, **457**, 854–858.

Wang,J. *et al*. (2017) POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. *Bioinformatics*, **33**, 2756–2758.

Wang,J. *et al*. (2018) Bastion6: a bioinformatics approach for accurate prediction of type VI secreted effectors. *Bioinformatics*, doi: 10.1093/bioinformatics/bty155.

Xiao,X. *et al*. (2013) iAMP-2L: a two-level multi-label classifier for identifying antimicrobial peptides and their functional types. *Anal. Biochem*., **436**, 168–177.

Xiao,X. *et al*. (2017) pLoc-mGpos: incorporate key gene ontology information into general PseAAC for predicting subcellular localization of Gram-positive bacterial proteins. *Nat. Sci*., **9**, 331–349.

Xu,Y. *et al*. (2013a) iSNO-PseAAC: predict cysteine S-nitrosylation sites in proteins by incorporating position specific amino acid propensity into pseudo amino acid composition. *PLoS ONE*, **8**, e55844.

Xu,Y. *et al*. (2013b) iSNO-AAPair: incorporating amino acid pairwise coupling into PseAAC for predicting cysteine S-nitrosylation sites in proteins. *PeerJ*, **1**, e171.

Xu,Y. *et al*. (2014) iNitro-Tyr: prediction of nitrotyrosine sites in proteins with general pseudo amino acid composition. *PLoS One*, **9**, e105018.

Xu,Y. *et al*. (2017) iPreny-PseAAC: identify C-terminal cysteine prenylation sites in proteins by incorporating two tiers of sequence couplings into PseAAC. *Med. Chem*., **13**, 544–551.

Yang,B. *et al*. (2017) BiRen: predicting enhancers with a deep-learning-based model using the DNA sequence alone. *Bioinformatics*, **33**, 1930–1936.

Yang,H. *et al*. (2018) iRSpot-Pse6NC: identifying recombination spots in Saccharomyces cerevisiae by incorporating hexamer composition into general PseKNC. *Int. J. Biol. Sci*., **14**, 883–891.

Yasser,E.M. *et al*. (2008) Predicting flexible length linear B-cell epitopes. *Computational Systems Bioinformatics*, **7**, 121–132.