Molecular Genetics and Genomics
https://doi.org/10.1007/s00438-018-1487-5

ORIGINAL ARTICLE

iPro70-FMWin: identifying Sigma70 promoters using multiple windowing and minimal features

Md. Siddiqur Rahman[1] · Usma Aktar[1] · Md Rafsan Jani[1] · Swakkhar Shatabda[1]

Received: 13 July 2018 / Accepted: 29 August 2018

Abstract

In bacterial DNA, there are specific sequences of nucleotides called promoters that can bind to the RNA polymerase. Sigma70 (σ^{70}) is one of the most important promoter sequences due to its presence in most of the DNA regulatory functions. In this paper, we identify the most effective and optimal sequence-based features for prediction of σ^{70} promoter sequences in a bacterial genome. We used both short-range and long-range DNA sequences in our proposed method. A very small number of effective features are selected from a large number of the extracted features using multi-window of different sizes within the DNA sequences. We call our prediction method iPro70-FMWin and made it freely accessible online via a web application established at http://ipro70.pythonanywhere.com/server for the sake of convenience of the researchers. We have tested our method using a standard benchmark dataset. In the experiments, iPro70-FMWin has achieved an area under the curve of the receiver operating characteristic and accuracy of 0.959 and 90.57%, respectively, which significantly outperforms the state-of-the-art predictors.

Keywords σ^{70} promoter · Prokaryote · Sequence-based features · Multi-windowing · Feature selection

Introduction

Promoters are binding sites in genes for RNA polymerase. Promoters are the sequence of several nucleotides, which are essential for the initialization and regulation of gene transcription. Several key factors are involved in the process of gene transcription. In bacteria, sigma (σ) factor is a subunit of RNA polymerase and one of the most contributor for recognizing and binding promoter sequence during gene transcription. There are several sigma (σ) factors for recognizing different promoters sequences, for instance, $\sigma^{19}, \sigma^{24}, \sigma^{28}, \sigma^{32}, \sigma^{38}, \sigma^{54}$ and σ^{70}. Among them, σ^{70} factor also called "housekeeping" sigma factor or primary sigma factor is responsible for the regulation of the transcription of most genes in growing cells (Gruber and Gross 2003). Specific sigma factor is responsible to regulate specific gene transcription. Therefore, promoter sequences are defined by the name of sigma factor.

With the growth of available promoter sequences from previous laboratory experiments, computational algorithms are becoming popular instead of wet experimental approach. This is due to the time consuming and expensive nature of laboratory methods (Towsey et al. 2008) compared to faster and cost-effective computational methods. A good number of machine-learning algorithms are used to develop computational tools for prediction methods in the literature. They include support vector machines (SVM) (Lin et al. 2014, 2017), artificial neural networks (ANN) (Demeler and Zhou 1991; Lukashin et al. 1989; e Silva et al. 2014), Markov models (Audic and Claverie 1997), hidden Markov models (Mallios et al. 2009) and random forests (Liu et al. 2017b). Apart from machine-learning methods, position weight matrices (Fickett and Hatzigeorgiou 1997) and phylogenetic foot-printing (Grech et al. 2007) are used for the prediction of promoter sequences.

Neural Networks was first used to predict promoter sequences by Demeler and Zhou (1991). Sequence alignment kernel-based σ^{70} promoter sequence predictor was proposed by Gordon et al. (2003). They trained their method on 683 experimentally validated *Escherichia coli* promoter

Communicated by S. Hohmann.

✉ Swakkhar Shatabda
swakkhar@cse.uiu.ac.bd

[1] Department of Computer Science and Engineering, United International University, Madani Avenue, Satarkul, Badda, Dhaka 1212, Bangladesh

Published online: 05 September 2018

sequences compared to only 80 promoter sequences trained by the former method. An analysis on σ^{70} sequences was performed by Li and Lin (2006). Li and Lin (2006) used position correlation scoring matrix in their work. A hybrid approach called IPMD (Lin and Li 2011) was developed for eukaryotic and prokaryotic promoter prediction using combination of position correlation score function and Mahalanobis discriminant. A feature subspace-based ensemble classifier was used to identify σ^{70} promoter sequence (Rahman et al. 2018b)

Pro54BD, an experimentally verified database was proposed by Liang et al. (2017) that contained 210 experimentally verified σ^{54} sequences. Lin et al. (2014) used support vector machines to predict σ^{54} promoter sequences and proposed iPro54-PseKNC. iPro70-PseZNC was proposed by Lin et al. (2017) for σ^{70} promoter sequence detection using support vector machine and pseudo-nucleotide composition. Liu et al. (2017b) proposed iPromoter-2L which is a two-layer promoter sequence detector. In the first layer, a classifier is used to identify the promoter sequences from non-promoter regions and then in the second layer it distinguishes between several types of sigma promoter sequences. They have used Random Forest Classifier to predict different sigma factors.

In this paper, we have used sequence-based features for prediction of sigma70 promoter sequences. The main motivation of this work to rely on the hypothesis that the information about the promoter sequences must be there within the DNA sequences. However, we have included a number of features extracted from the sequences that include different statistical measures, position-specific counts, k-mer compositions, gapped compositions, etc. In addition to these, we have applied multiple window-based feature extraction to enhance the features. We have performed an analysis on the features to select an optimal set of features. Based on the optimal set of features found in our experiments, we have proposed a novel predictor named iPro70-FMWin. We have used a standard benchmark dataset to test the performance of iPro70-FMWin. Our proposed method significantly outperforms the state-of-the-art methods in terms of standard statistical measures. For the practical testing purpose, a webserver was developed for the identification of σ^{70} promoter sequences and can be freely accessible at http://ipro70.pythonanywhere.com/server.

Materials and methods

In this section, we present the methodology of the paper as suggested by Chou as five steps for developing methods for prediction of attributes of biological entities (Chou 2011a, 2013). These five steps are: (1) the description of the benchmark dataset use; (2) representation of the sample or instances; (3) description of the classification algorithms; (4) performance evaluation techniques and last, (5) development of a web server. These suggestion is followed in many of the work in the literature (Yang et al. 2018; Su et al. 2018; Liu et al. 2018a, b; Feng et al. 2018; Chen et al. 2018). The rest of the section is organized in the similar fashion.

Benchmark dataset

The first step in developing a prediction method is to construct a benchmark dataset for the classification problem. In this work, a high-quality pre-constructed benchmark dataset was collected from a reliable data source: http://lin-group.cn/server/iPro70PseZNC/data.html (Lin et al. 2017). The raw data have been collected from RegulonDB 9.0 (http://regulondb.ccg.unam.mx/; Gama-Castro et al. 2015). There were total 2141 DNA sequences. We have used cross-validation on those total dataset to train and test our method. Among them 741 were σ^{70} promoter sequences of *E. coli* K-12. Rest of them were randomly chosen non-promoter sequences which were extracted from coding regions and intergenic regions of *E. coli* K-12 genome. All of them were 81 bp where 60 bp upstream and 20 bp downstream of the TSS (Transcript Start Site) at each sequence in the dataset. The constructed dataset was slightly imbalanced among promoters and non-promoters (ratio 1:1.89) and which can be expressed by

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

where $\mathbb{S}^{+}$ represents positive samples or promoter DNA sequences, $\mathbb{S}^{-}$ represents negative samples or non-promoter DNA sequences, and the symbol $\cup$ represents the union in the set theory.

Formulation of DNA samples

A DNA sequence is the combination of four nucleic acids named adenine (A), guanine (G), cytosine (C), and thymine (T). Since a machine-learning algorithm can recognize only numerical features as input; therefore, we needed to formulate the DNA sequences with an effective mathematical illustration for feature extraction. The straightforward representation of a DNA segment with its entire nucleic acid residues can be formulated as

$$D = R_1R_2R_3R_4R_5R_6 \dots R_L, \tag{2}$$

where each nucleic acid residue is represented as $R_{i(1,2,3,4,\dots,L)}$ at the position of i in the DNA sequence, where L the length of that DNA sequence. Typically, the input requirement of a machine-learning algorithm is a numerical representation of objects for the statistical prediction analysis. In this post-genomic era, there has been an explosive growth in biological sequences and its a challenge to express the sequences as vectors as required for machine-learning algorithms (Chou 2015). Now, the first step is how can we express DNA sequence into a numerical representation as an input for the machine-learning algorithm to get as much as a possible accuracy of prediction.

In last two decades, many exciting methods have been invented which has already done a thrilling result in predicting promoter sequences (Lin et al. 2014, 2017; Gan et al. 2012; Shin and Noireaux 2010). One of the most successful vector representation is the pseudo-amino acid composition (Chou 2001a) or PseAAC (Chou 2004). It has been widely used in different areas of computational proteomics (Arif et al. 2018; Mei and Zhao 2018a, b; Krishnan 2018; Zhang and Duan 2018; Contreras-Torres 2018; Rahman et al. 2018a; Sabooh et al. 2018; Chou 2017). Encouraged by the successes of using PseAAC (Chou 2009, 2011b) to deal with protein/peptide sequences, the idea of PseAAC was extended to PseKNC (pseudo K-tuple nucleotide composition) (Chen et al. 2014) to generate various feature vectors for DNA/RNA sequences that have proved very successful (Yang et al. 2018; Su et al. 2018; Liu et al. 2018a, b; Feng et al. 2018; Chen et al. 2015, 2018). Pse-in-One (Liu et al. 2015) and Pse-in-One 2.0 (Liu et al. 2017a) webservers are able to generate any desired feature vectors for protein/peptide and DNA/RNA sequences.

In this work, we focused on multiple window approach to extract features. Detail about feature extraction is shown in bellow.

Statistical measures

Among the statistical measures, we first included a simple frequency count of each nucleotide along through the 81-bp DNA segment. One of the other important features is the GC content (Yamagishi 1974) with a high ranked promoter-predicting score. We have also incorporated three common statistical measures. Those are standard deviation, mean, and variance. Low standard deviation indicates that the data distribution is closed to the mean of the distribution, whereas the high standard deviation indicates spreading range of the data distribution. In addition, according to the previous research, there should be an equal frequency of the four DNA bases (A, C, G, T) (Lobry 1996) if there is no mutational or selective pressure. Therefore, we also used GC-skew (Ginno et al. 2013) for the whole sequence.

k-mer composition

k-mer stands for the *k*-length substring of all possible combination of A, C, G, and T through the full string of the given DNA segment. *k*-mer is effectively used in the field of computational biologies such as sequence assembly, sequence alignment, and variability in human genome mutation rate explanation (Compeau et al. 2011; Samocha et al. 2014; Aggarwala and Voight 2015). In this work, we constructed features using *k*-mer, where $k = 2, 3, 4, 5, 6$. The frequency or composition of the *k*-mer was taken as features. These features have been successfully used in the literature for nucleosome position prediction (Guo et al. 2014), recombination spot prediction (Yang et al. 2018), RNA subcellular localization (Su et al. 2018), bacterial σ^{70} promoter prediction (Rahman et al. 2018b) and origin of replication prediction (Zhang et al. 2016).

g-gapped *k*-mer composition

In addition to *k*-mer compositions, we have used *g*-gapped dinucleotide compositions. However, we extend the idea of dinucleotides to tri-nucleotides as well and hence towards a possible generalization of any *k*-mers. We have considered gaps, $g = 1, 2, 3, \ldots, 24$. More detail about this feature is shown in Table 1.

When constructing the features for *k*-mers, $k = 3$, we used patterns such as X_XX and XX_X. Note that, gapped *k*-mers were first proposed for solving the protein subcellular localization prediction problem (Li and Li 2008) and widely used for protein function prediction (Tang et al. 2016, 2018; Chen et al. 2016; Yang et al. 2016).

Pattern finding

It has been proven that there are some patterns of nucleotide sequences which are very important to recognize promoter region in DNA sequence (Huerta and Collado-Vides 2003; Olson et al. 2015; Stormo 2000). Some of them have been used in this work for feature extraction such as TATAAT, TAATAT, TATAAA, AAATAT, TTGACA, ACAGTT, and AACGAT. We tried to exact or approximate pattern matching for all of them and also their right-shifted cyclic form except the last one. The string matching of each pattern has been counted if there was at least three matches. Otherwise, the counted value was considered as a zero.

Table 1 *k*-mer composition using *g*-gap

k	*g*	Pattern
2	1	X_X
2	2	X__X
2	3	X___X
…	…	… … …
…	…	… … …
2	24	X________________________X

Positioning distance count

In this paper, we also consider the total summation of each nucleotide positioning distance. The calculation of this feature is illustrated in Fig. 1. For this example, total summation of positioning distance for A, C, G, and T are 9, 4, 4, and 4, respectively.

Identification of promoter regulatory regions can be effectively found with a high prediction scores by another popular approach called DNase I hypersensitive site sequencing (El Hassan and Calladine 1996; Crawford et al. 2006; Boyle et al. 2008). Similarly, we have used dinucleotide parameters based on DNaseI digestion data in this article.

Multiple window approach

Typically, most of the previous researchers have used the full length of the given DNA segment for feature extraction. In this study, window method has been applied to the above-described all the feature group to get an extra benefit. However, different window size was for the different feature group. Details of the window approach are shown in Table 2. A windowing method for prediction of origin of replication is done in Liu et al. (2018b).

Feature selection

The feature selection strategy typically chooses a small number of useful features from a large number of unnecessary or irrelevant features. This is broadly used to achieve the most important and short-sized subsets of total features (Dash and Liu 1997). Practical experiences have proven that more effective prediction tools are achievable by feature selection. In our study, we have implemented the feature selection strategy for the following four aspects: (1) model simplification (James et al. 2013), (2) reduce training time, (3) avoid dimensionality problem, and (4) reduce over-fitting (Bermingham et al. 2015). To full-fill those four aspects, we have used the AdaBoost algorithm to find the best features based on their predictive accuracy.

AdaBoost training for feature selection

To improve prediction performance, the AdaBoost algorithm is often used with other types of machine-learning algorithms. Initially, a simple classifier has been fitted on the data also called a decision stump which splits the data into just two regions. Then, the class correctly classified will be given less weight edge in the next iteration and higher weight edge from misclassified classes. After that, another decision stump or weak classifier will be fitted on the data and will change the weights again for the next iteration. Here check the misclassified for which weight has been increased once it finishes the iterations these are combined with weights. Also, weights are automatically calculated for each classifier at each iteration based on error rate to come up with a strong classifier which predicts the classes with surprising accuracy. Let us have a look at the equation for AdaBoost,

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \lambda_j \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where $h_j(x)$ is a simple threshold function consisting of only one simple feature $f_j(x)$, λ_j is a threshold and p_j is a parity to indicate the direction of the inequality. The threshold value is determined by the mean value of the positive samples and the mean value of the negative samples on the jth feature response:

$$\lambda_j = \frac{1}{2}\left(\frac{1}{m}\sum_{p=1}^{m} f_j(x_{\mathrm{p}} | y_{\mathrm{p}} = 1) + \frac{1}{l}\sum_{n=1}^{l} f_j(x_{\mathrm{n}} | y_{\mathrm{m}} = 0)\right). \tag{4}$$

More details about this procedure are to be found in the work by Shen and Bai (2004). In this study, we have performed the AdaBoost algorithm to identify the features which have most prediction capability. Finally, we have found a small number of effective features with a great accuracy. To find the best feature set, we applied AdaBoost using tenfold cross-validation on our feature set. Then, it selected different features in each fold as the best features. We have just tried to count their frequency based on their presence in each fold. After, we have found only 27 common features among 22,595 features (shown in Table 2) which were present in all tenfolds. Then, we trained our model by only those 27 features and test by the new test set.

Classification algorithms

Choosing classifiers or model is crucial to get a desirable outcome. One of the main focusing points in this work was

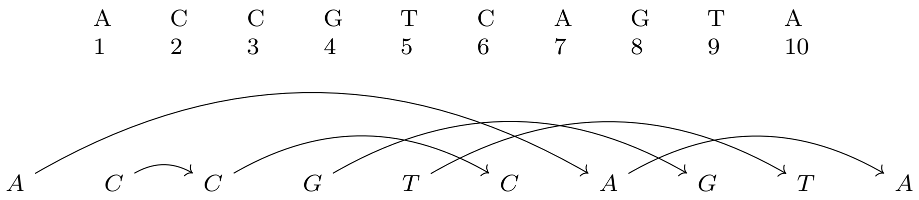

Fig. 1 Illustration of positioning distance count feature

Table 2 Applying windowing on all the above-described features

Group	Feature name	Window size (bp)	Total
1	Frequency count of each nucleotides and (G+C) content	− 60 to +20	5
		− 60 to − 51	5
		− 50 to − 41	5
		− 40 to − 26	5
		− 15 to − 6	5
		− 5 to − 1	5
		0 to − 9	5
		10 to 20	5
2	Mean, variance, standard deviation	− 60 to 20	3
		− 15 to − 6	3
3	GC-skew	− 60 to 20	81
4	k-mer, $k = 2, 3, 4, 5, 6$	− 60 to 20	5456
	$N^2 + N^3 + N^4 + N^5 + N^6 = 5456$	− 40 to − 26	5456
	$N = 4$ for A, C, G, and T	− 15 to − 6	5456
5	g-gap into k-mer, $g = 1, 2, 3,\ldots, 24$ and $k = 2, 16 \times 24 = 384$	− 60 to 20	384
	g-gap into k-mer, $g = 1, 2, 3,\ldots, 11$ and $k = 2, 16 \times 11 = 176$	− 40 to − 26	176
	g-gap into k-mer, $g = 1, 2, 3,\ldots, 7$ and $k = 2, 16 \times 7 = 112$	− 15 to − 6	112
6	g-gap into k-mer, $g = 1, 2, 3,\ldots, 24$ and $k = 3, 64 \times 24 \times 2 = 3072$	− 60 to 20	3072
	g-gap into k-mer, $g = 1, 2, 3,\ldots, 11$ and $k = 3, 64 \times 11 \times 2 = 1408$	− 40 to − 26	1408
	g-gap into k-mer, $g = 1, 2, 3,\ldots, 7$ and $k = 3, 64 \times 7 \times 2 = 896$	− 15 to − 6	896
7	Approximate cyclic right-shifted pattern count:	− 15 to − 6	24
	TATAAT, TAATAT, TATAAA, AAATAT, $6 \times 4 = 24$		
8	Approximate cyclic right-shifted pattern count:	− 40 to − 26	12
	TTGACA and ACAGTT, $6 \times 2 = 12$		
9	Approximate pattern count: AACGAT	1 to 6	1
10	Positioning distance count for A, C, G, and T	− 60 to 20	4
		− 40 to − 26	4
		− 15 to − 6	4
11	Dinucleotide parameters based on DNaseI digestion data	− 60 to 20	1
		− 40 to − 26	1
		− 15 to − 6	1
			Total: 22,595

to choose an effective classification model for our supervised learning task. For picking up the perfect classifier, we tried different classifiers on our sample data such as support vector machine (SVM) (Cortes and Vapnik 1995), logistic regression (LR) (Hosmer Jr et al. 2013), K-nearest neighbor (KNN) (Altman 1992), decision tree classifier (DTC) (Safavian and Landgrebe 1991), Gaussian Naive Bayes (GNB) (Murphy 2006) and linear discriminant analysis (LDA) (Mika et al. 1999). After that, we summarized them in one classifier. In this section, we will discuss a brief on them.

Support vector machine (SVM) (Cortes and Vapnik 1995) is a supervised learning classifier that tries to maximize the margin between two classes by mapping input data instances to a higher dimensional space. In other words, it looks at the extremes of the dataset and draws a decision boundary also known as a hyperplane near the extreme points in the dataset. We performed classification by finding the hyperplane that differentiates the two classes. Essentially, kernel function has been used to transform nonlinear into a linear space. There are some popular kernel functions to transform data into high-dimensional feature space such as polynomial kernel, radial basis function, and sigmoid kernel. Choosing the correct kernel function is a non-trivial task. A popular parameter-choosing technique k-fold cross-validation has been used to choose our parameter. We used the radial basis function (RBF) kernel to overcome the problem with transformation into higher dimensional feature space. The downside of SVM is the training time much longer as its much more computationally intensive.

Linear discriminant analysis (LDA) (Mika et al. 1999) is most commonly used as a dimensionality reduction technique in the preprocessing step for data classification in machine-learning applications. The goal is to project a dataset into a lower dimensional space with good class separability to avoid over-fitting. Logistic regression (LR) (Hosmer Jr et al. 2013) is another popular technique borrowed by machine learning from the field statistics. Logistic regression is similar to linear regression in a sense that they both have the same goal of estimating the values of parameters coefficients. As a result, at the end of the training of the machine-learning model, we got the function that best describes the relationship between the non-input and the output values, unlike linear regression the prediction of the output is transformed using a nonlinear function called the logistic function. Some variations of the same are called the sigmoid function as well as the logit function.

In pattern recognition, K-nearest neighbor (KNN) (Altman 1992) algorithm is a method for classifying the object based on the closest training example in the feature space. KNN is a type of instance-based learning where the function is only approximated locally and all competition is delayed until classification. The KNN algorithm is fundamental and one of the simplest classification technique when they little or no prior knowledge about the distribution of the data. The K in KNN refers to the number of nearest neighbors that the classifier will use to make its prediction. Another type of supervised machine-learning algorithm is a decision tree classifier (DTC) (Safavian and Landgrebe 1991). A tree has many analogies in life and turns out it is influenced by a wide area of machine learning covering both classification and regression trees, otherwise known as caught. A decision tree is a structure where each internal node denotes a test on an attribute each branch represents an outcome of a test and each leaf or terminal node holds a class label. The topmost node in a tree is the root node. In decision analysis, a decision tree can be used to individually and explicitly represent decisions and decision making as the name it uses a tree, like a model of decisions. So the advantages of decision tree are: (1) it is simple to understand, interpret and visualize, (2) decision tree implicitly performs variable screening or feature selection, (3) it can handle both numerical as well as categorical data, (4) it can also handle multi-output problems; decision trees requires relatively little effort from the user for data preparation and (5) nonlinear relationships between parameters do not affect the clip performance. The disadvantage of cost, however, is that the decision tree learns to create over complex trees that do not generalize the data well. This is also known as over-fitting.

Gaussian Naive Bayes (GNB) (Murphy 2006) is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model".

Performance evaluation

We have used cross-validation sampling technique to tune the parameters of our classification algorithm, feature selection and evaluate the performance of the predictor. Cross-validation (Kohavi et al. 1995) is a robust method for sampling instances and widely used in the literature. In k-fold cross-validation, sometimes called rotational estimation, the dataset D is randomly divided into k subsets or folds ($D_1, D_2, \dots, D_k$) of approximately equal size. The algorithms are trained and tested k times; each time $t \in 1, 2, \dots, k$, it is trained on $D \backslash D_t$ and tested on D_t. The cross-validation estimated accuracy is the overall number of correct classifications, divided by the number of instances in the dataset. Formally, let $D_{(i)}$ be the test set that includes instance $x_i = \langle v_i, y_i \rangle$, then the cross-validation estimate of accuracy,

$$\mathrm{acc}_{cv} = \frac{1}{n} \sum_{\langle v_i, y_i \rangle \in D} \delta(L(D \backslash D_{(i)}, v_i)). \tag{5}$$

According to k-fold cross-validation, to train and test our data, we have tried k-fold cross-validation for the six popular machine-learning algorithms described in the previous section, where $k = 2, 5, 10, 20$. To select appropriate parameters for each classifier, we used grid search (Coussement and Van den Poel 2008) on our sample data using k-fold cross-validation, where also $k = 2, 5, 10, 20$. By doing different experiments, we have found the best parameters for each of those methods. In this study, we have tried all those machine-learning algorithms from Scikit-learn library (Pedregosa et al. 2011) in python programming.

In this paper, we have mostly used seven measures to evaluate the performance of the algorithms for σ^{70} promoter prediction. They are accuracy (Acc), sensitivity (S_n), specificity (S_p), $F1$-score, area under precision recall curve (auPR), area under receiver operating characteristic curve (auROC), and Matthew's correlation coefficient (MCC). For any two-label classification problem accuracy (Acc) can be represented formally.

$$\mathrm{Acc} = \frac{\mathrm{TN} + \mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{TN} + \mathrm{FP}}. \tag{6}$$

Here, TP represents true positive, that is the total number of correctly classified positive instances or predicted class type is positive and actual class type is positive. TN represents true negative or the total number, that is correctly classified negative instances or predicted class type is negative and actual class type is negative. FP represents false positive means that predicted class type is positive but the actual class type is negative. FN represents false negative which means that the predicted class type is negative but the actual class type is positive. The value of accuracy ranges between 0 and 100%.

Sensitivity (S_{n}) or recall is another important metric which is the true positive rate or number of correctly classified positive instances over the total number of positive examples that can be formally defined as follows:

$$S_{\mathrm{n}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}. \tag{7}$$

Similar to specificity (S_{p}) which is the true negative rate or number of correctly classified negative instances over the total number of negative examples that can be formally defined as follows:

$$S_{\mathrm{p}} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}. \tag{8}$$

$F1$-score is the harmonic mean of precision and recall. It is defined as:

$$F_1\text{-Score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{9}$$

where *precision* is defined by the ratio of the number of true positives over the total number of positive predictions.

$$\text{precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}. \tag{10}$$

All these measures have the range of values in {0,1}. A higher value of these metrics indicates a better performing predictor.

Matthew's correlation coefficient (MCC) is another measure approach to evaluate the model. The range of values is in $[-1, 1]$ which, respectively, denotes negative classification correlation and positive classification correlation. We can calculate MCC directly from confusion matrix using the following formula:

$$\mathrm{MCC} = \frac{(\mathrm{TP}\times\mathrm{TN}) - (\mathrm{FP}\times\mathrm{FN})}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}. \tag{11}$$

Area under the precision recall curve (auPR) and area under the receiver operating characteristic curve (auROC) are important measures. They reveal the strength of the underlying classifier regardless of the threshold chosen. Note that, Chou introduced the similar metrics using a different set of symbols (Chou 2001b, c; Chen et al. 2013) and followed by a number of work in the literature (Rahman et al. 2018a; Chen et al. 2018; Feng et al. 2018).

Using these measures, it is very important to choose sampling techniques for classification algorithms. Most common are independent test sets and cross-validations. In this paper, we have used cross-validation since they are robust, reduces over-fitting and widely used in the literature of sigma promoters prediction and thus suitable for comparison of different algorithms.

Results

For experimental purposes, we have implemented all the methods using python programming for data analysis and statistical testing. We also have used the Scikit-learn library (Pedregosa et al. 2011) where most of the machine-learning algorithms are available. In addition, we have plotted Boxplot (Williamson et al. 1989) and receiver operating characteristic (ROC) curve for visually comparing the experimental results.

There were three major experiments done in our work. For the first experiments, we have extracted all our features based on 81-bp full-length inputted DNA segments. All the sequences were randomly divided into two sets: one was the training data set and another one was the test data set. The test set was completely independent from the training set in each analysis. The dataset was divided into k equal parts for k-fold cross-validation. In this case, $k-1$ parts were used for training and the kth part was used for testing. In tenfold cross-validation, the procedure was continuing ten times in the manner of rotation. The process is depicted in Fig. 2. Within the tenfold cross-validation, we have applied six popular machine-learning algorithms. They are support

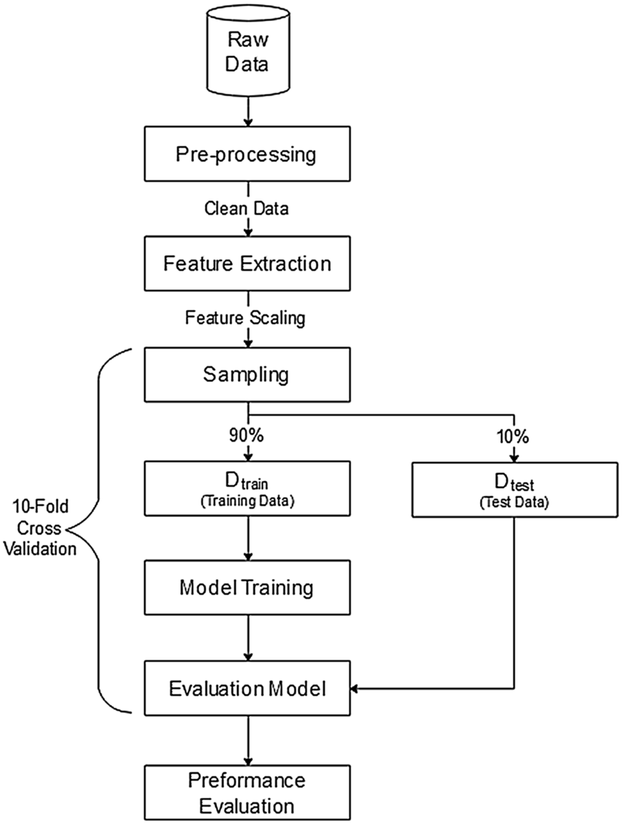

Fig. 2 Showing typical classification approach

Table 3 Experimental results for the approach in Fig. 2

Classifiers	Acc (%)	auROC	auPR	S_p (%)	S_n (%)	MCC	F_1
SVM	**82.81**	**0.901**	0.829	89.93	69.37	0.612	0.735
LR	81.64	0.882	0.804	81.57	81.78	0.614	0.755
KNN	75.90	0.812	0.646	82.21	63.97	0.465	0.648
DTC	71.18	0.681	0.488	78.07	58.16	0.363	0.581
GNB	74.54	0.764	0.569	74.29	75.03	0.474	0.671
LDA	73.52	0.787	0.668	78.93	63.29	0.419	0.623

Bold values indicate the best values

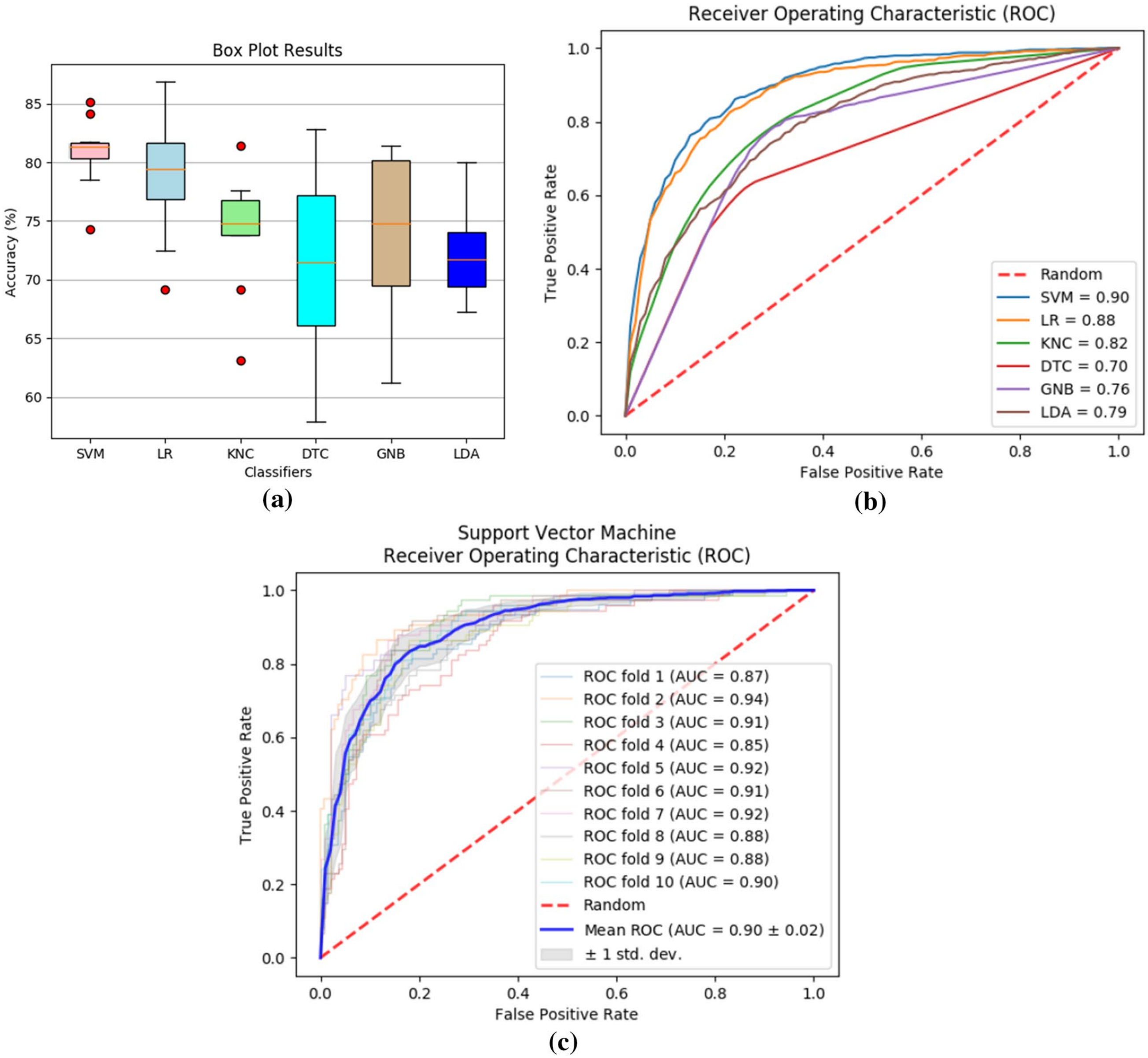

Fig. 3 Graphical depiction of performance of the method of Fig. 2: **a** box-plot of different classifiers on the dataset among cross-folds; **b** receiver operating characteristic curve of different classifiers; **c** receiver operating characteristic curve of different folds in the cross-validation of SVM, the best performing classifier

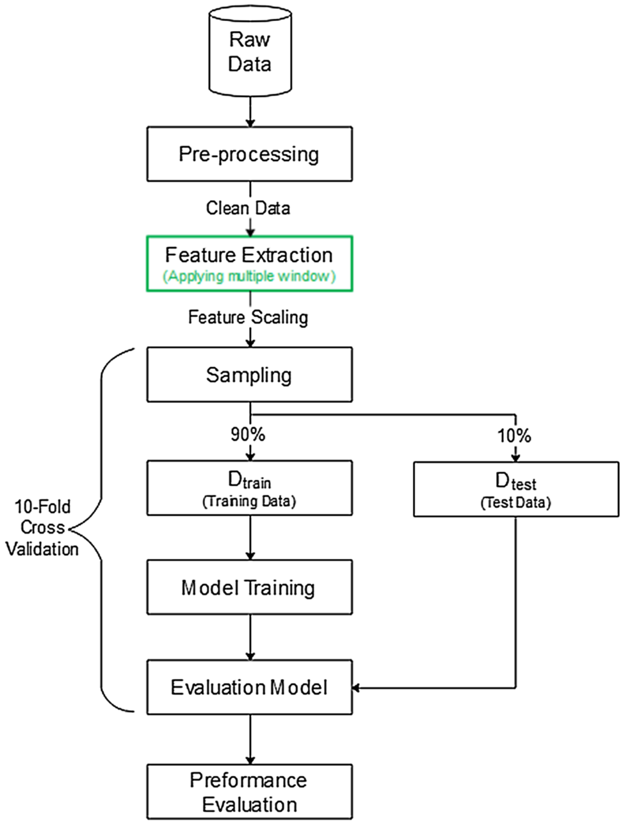

Fig. 4 Applying window to feature extraction

vector machines (SVM) (Cortes and Vapnik 1995), logistic regression (LR) (Hosmer Jr et al. 2013), K-nearest neighbor (KNN) (Altman 1992), decision tree classifier (DTC) (Safavian and Landgrebe 1991), Gaussian Naive Bayes (GNB) (Murphy 2006) and linear discriminant analysis (LDA) (Mika et al. 1999). We report the experimental results achieved by these six algorithms in Table 3. From the results obtained in these experiments, it is clear that the best performing method is SVM. Logistic regression is also working very nicely on this set of all features and the performances are very close to that of SVM. To illustrate the performance of the algorithms in a better way, a box-plot of accuracy of different algorithms in Fig. 3a and ROC curve (Hanley and McNeil 1982) in Fig. 3b on the decisions provided by those six classifiers are presented. Please note that box-plot shows a few outliers (Hodge and Austin 2004). After that, we have picked up the best classifier among them based on their different scores such as Acc, auROC, auPR, S_p, S_p, MCC and F_1 Score (Table 3). In this case, it was SVM. To see the decision fluctuation, we have plotted the ROC in Fig. 3c for this best classifier SVM with the changes in its tenfold cross-validation for each fold.

Second, we have applied the same procedure, but in this case, we also have considered the short rage of DNA segments for feature extraction formulated by multiple windowing approaches and also including the full 81-bp-long sequence. Window size was tuned for different experiments. Then, we have run those previous six classifiers on these new feature group using tenfold cross-validation. Figure 4 depicts the idea of the set of experiments performed at this stage. This time we got a significant improvement in the results as shown in Table 4.

From the results reported in Table 4, we can note that the best performing classifiers were again logistic regression and SVM, where latter was slightly inferior in comparison. Box-plots shown in Fig. 5a reveal that there were no outliers after using multiple windowing approaches. We also plotted again ROC for all classifiers in Fig. 5b and we have found less fluctuation for the best classifier among the folds as shown in Fig. 5c compared to the previous set of experiments.

Finally, we have tried to find the minimal set of features in terms of their presence within all tenfold cross-validation from feature selection by AdaBoost algorithm (Shen and Bai 2004) (Table 5). The idea of the experiments is depicted in Fig. 6. Surprisingly, we have noticed that only 27 features are enough to lead most of our classifiers to predict the data with a high accuracy. Figure 8 shows individual feature's accuracy scores for all the classifiers used in this study. All of the classifiers did great except KNN. After that, once more we have trained our classifiers only by those 27 (Table 6) features and test the data. Now, we have seen an increase in results again as shown in Table 5. Also note the smooth box-plot and ROC curves seen in Fig. 7a, b and an acceptable fluctuation in ROC Fig. 7c for logistic regression,

Table 4 Experimental results for the approach in Fig. 4

Classifiers	Acc (%)	auROC	auPR	S_p (%)	S_n (%)	MCC	F_1
SVM	88.70	0.962	0.937	97.07	72.87	0.748	0.816
LR	**89.54**	**0.966**	0.949	87.71	92.98	0.783	0.862
KNN	70.95	0.829	0.670	63.00	85.96	0.467	0.672
DTC	82.39	0.803	0.644	87.00	73.68	0.609	0.742
GNB	78.42	0.713	0.577	94.50	48.04	0.504	0.605
LDA	83.05	0.889	0.831	87.86	73.95	0.623	0.752

Bold values indicate the best values

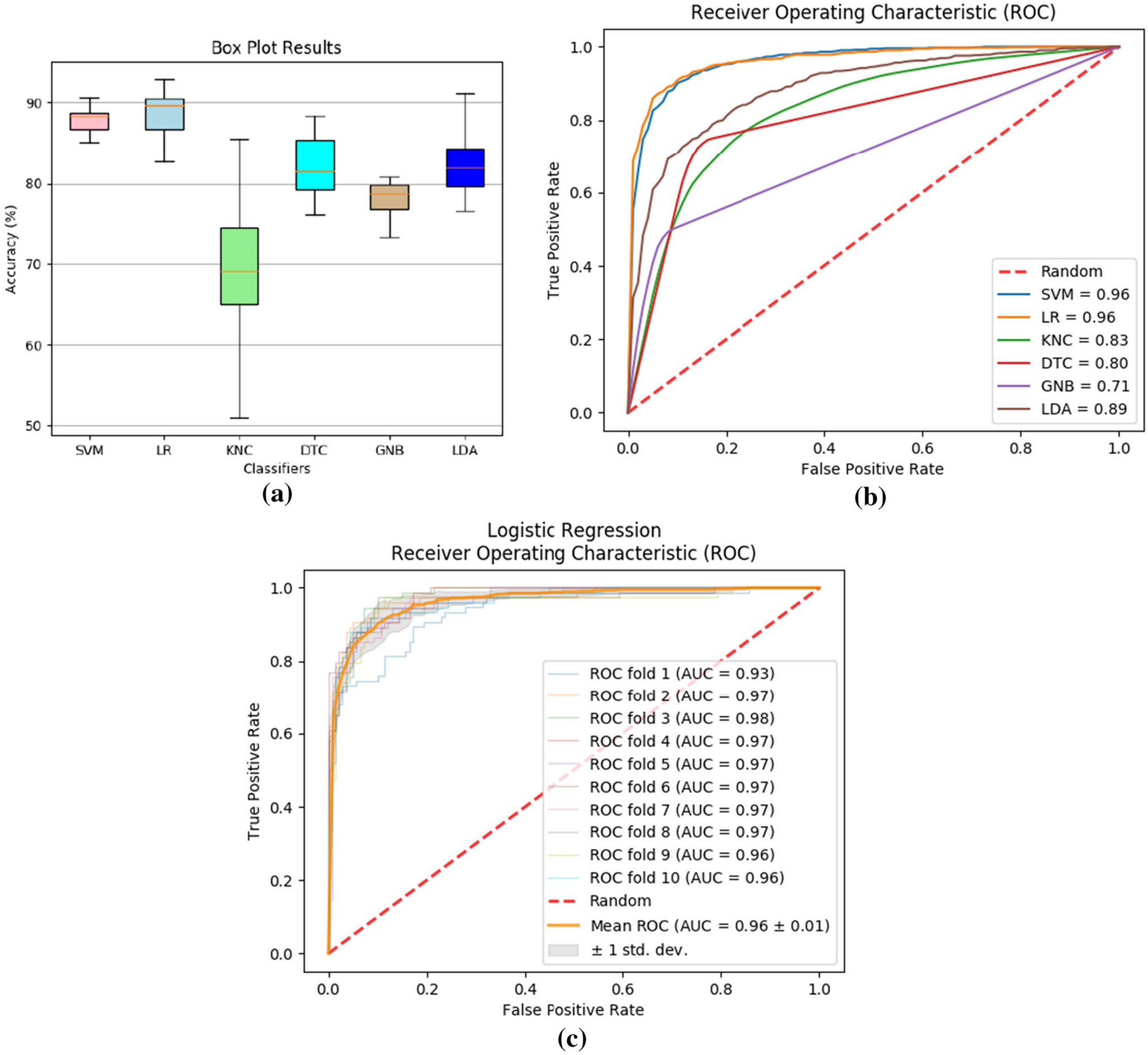

Fig. 5 Graphical depiction of performance of the method of Fig. 4: **a** box-plot of different classifiers on the dataset among cross-folds; **b** receiver operating characteristic curve of different classifiers; **c** receiver operating characteristic curve of different folds in the cross-validation of logistic regression, the best performing classifier

Table 5 Results for the approach in Fig. 6

Classifiers	Acc (%)	auROC	auPR	S_p (%)	S_n (%)	MCC	F_1
SVM	89.96	0.951	0.922	93.57	83.13	0.776	0.851
LR	**90.57**	**0.959**	0.937	94.43	83.27	0.789	0.859
KNN	88.32	0.922	0.870	95.36	75.03	0.738	0.816
DTC	82.67	0.810	0.653	86.50	75.44	0.618	0.751
GNB	88.28	0.952	0.912	88.79	87.31	0.748	0.839
LDA	90.24	0.958	0.938	96.00	79.35	0.782	0.849

Bold values indicate the best values

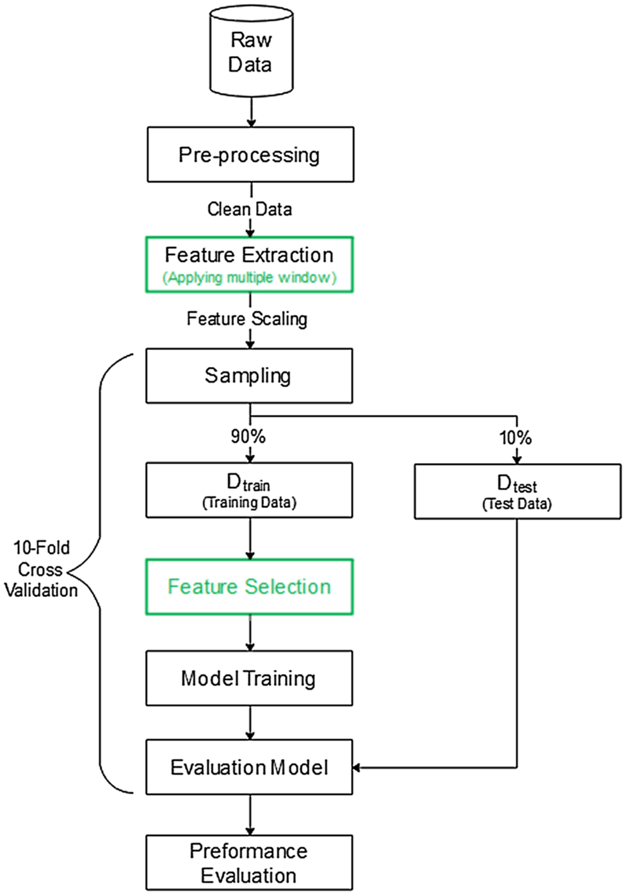

Fig. 6 After feature selection on both short-range and long-range DNA segments

Table 6 Features were found in frequency 10 in tenfold cross-validation after feature selection

Feature no.	Feature index	Pattern	Window size (bp)
4	F1	G+C	− 60 to + 20
24	F2	G+C	− 15 to − 6
105	F3	GC-skew	− 1 to − 1
107	F4	GC-skew	1 to 1
171	F5	TA	− 15 to − 6
237	F6	TTG	− 60 to 20
267	F7	CTA	− 40 to − 26
3152	F8	TTGAC	− 40 to − 26
16628	F9	C<– 9 gap –>C	− 60 to 20
17055	F10	A<– 1 gap –>A	− 15 to − 6
17375	F11	CA<– 4 gap –>A	− 60 to 20
17458	F12	GA<– 5 gap –>T	− 60 to 20
17602	F13	TA<– 7 gap –>T	− 60 to 20
17886	F14	AT<– 12 gap –>T	− 60 to 20
18510	F15	TT<– 21 gap –>T	− 60 to 20
18890	F16	TG<– 3 gap –>T	− 40 to − 26
19188	F17	GC<– 8 gap –>C	− 40 to − 26
19586	F18	TA<– 3 gap –>T	− 15 to − 6
19993	F19	A<– 3 gap –>GG	− 60 to 20
20492	F20	T<– 10 gap –>TC	− 60 to 20
20826	F21	A<– 16 gap –>GT	− 60 to 20
22162	F22	A<– 2 gap –>AT	− 15 to − 6
22166	F23	A<– 2 gap –>CT	− 15 to − 6
22543	F24	TATAAT	− 15 to − 6
22563	F25	TTATAA	− 15 to − 6
22567	F26	TTGACA	− 40 to − 26
22594	F27	DNaseI	− 15 to − 6

the best performing classifier in this case. Therefore, based on the overall accuracy and other measures found among all these experiments, we have selected Logistic Regression as a decision algorithm in our sequence-based predictor named “iPro70-FMWin” since it had the highest accuracy from logistic regression.

In this work, our feature selection method picked only 27 best features based on their frequency in all tenfolds. For the inquisitiveness, we also examined all classifiers for the other features which had the frequency of less than 10 in tenfolds. All the scores were increased gradually while we were considering the features by decreasing frequency restriction one by one such as $F = 9, 8, 7, \ldots, 1$ (Table 7). As it can be seen at Table 7, the number of features was increased while we considered frequencies by decreasing manner. According to the goal of our feature selection method, we have tried to optimize the features as less as possible. However, we agree that many previous works have done great using some special pattern of nucleotides such as TATAAT, TTA TAA, and TTGACA within specific regions, we also have taken into account those patterns exactly or partially in this work. But it can be seen that from Table 6 and Fig. 8, there were a great influence of *g*-gap features in terms of identifying σ^{70} promoter sequence. According to Fig. 8, the most effective features in this work is frequency count of “F18” which is TAXXXT (X = A, C, G or T) within − 15 to − 6 bp in Table 6. We also found only “F5” is an important motif among our 27 in Fig. 8 in a sense where all our classifiers were aligned based on their accuracy; but for the other case, most of their curve was overlapped except KNN. It was happiest to see that most of the features were playing an effective role for all classifiers except KNN.

Comparison with previous methods

A large number of features were used in many previous methods (Lin et al. 2014, 2017; Gordon et al. 2005) to recognize promoter sequence. While the goal of this research was to optimize the number of features into a

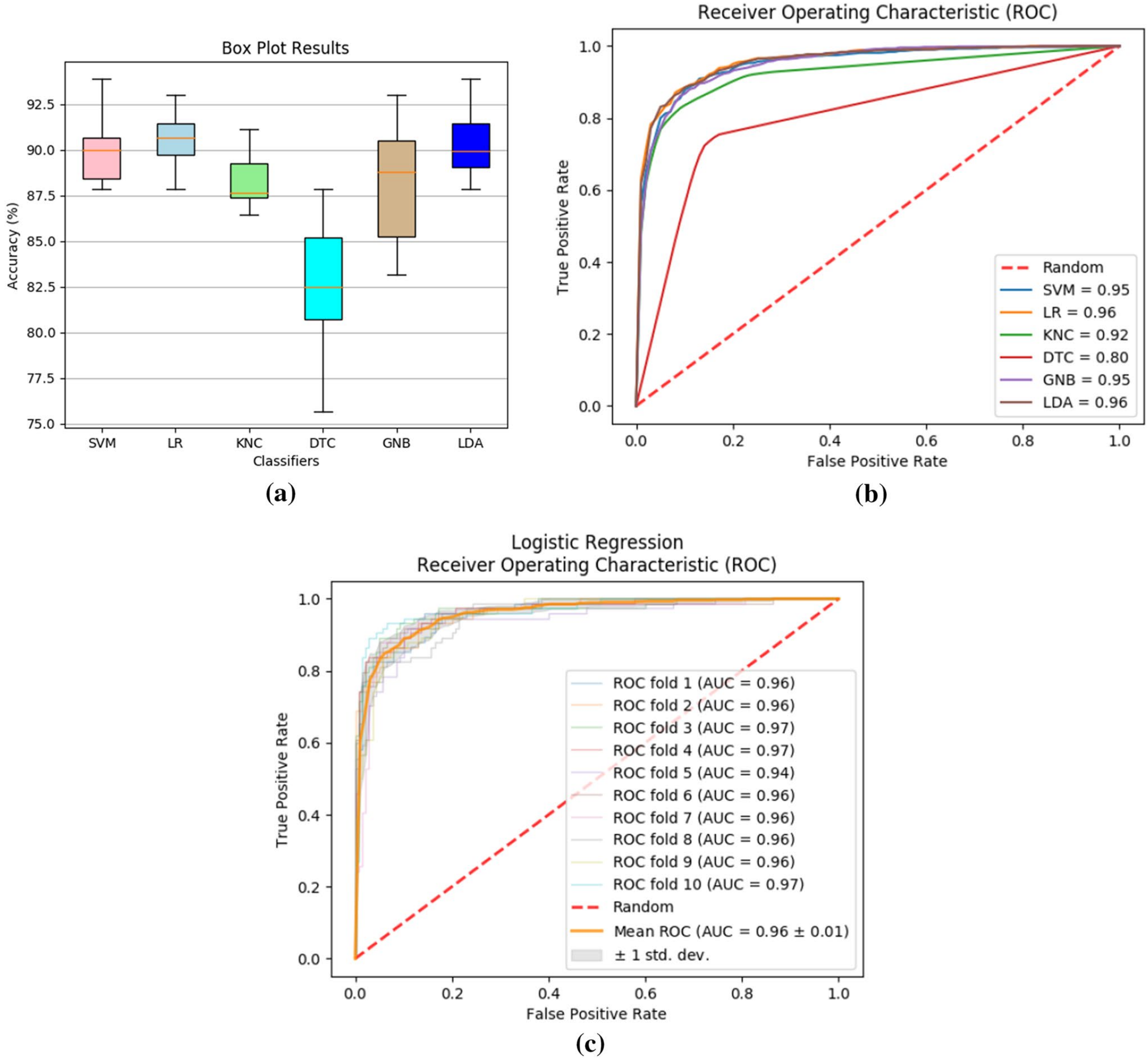

Fig. 7 Graphical depiction of performance of the method of Fig. 6: **a** box-plot of different classifiers on the dataset among cross-folds; **b** receiver operating characteristic curve of different classifiers; **c** receiver operating characteristic curve of different folds in the cross-validation of logistic regression, the best performing classifier

small number of effective features. However, we have compared the performance of our algorithm with two other state-of-the-art algorithms. The experimental report of iPro70-FMWin was recorded in contrast with three other popular methods: iPro70-PseZNC (Lin et al. 2017), IPMD (Lin and Li 2011) and Z-curve (Song 2011) in Table 8.

From the values reported in Table 8, it could be noted that iPro70-FMWin shows significantly improved performance on all the metrics except sensitivity, where it is very close in performance compared to IPMD.

Web-server implementation

It is a very important step to develop a user-friendly and easy-to-access web application for useful prediction methods. This is suggested in Chou and Shen (2009) and demonstrated in a series of recent publications (Liu et al. 2018a, b; Rahman et al. 2018a; Rayhan et al. 2017). We have designed our feature extraction model in python programming only for those 27 effective features. We also have implemented logistic regression for classification in

Table 7 Feature selection with their frequency within tenfolds and result of the best classifier for the corresponding features

No. of features	Frequency	Best classifier	Acc	auROC	auPR	S_p	S_n	MCC	F_1
27	10	LR	90.57	0.959	0.937	94.43	83.27	0.789	0.859
55	9	LR	91.78	0.973	0.955	94.71	86.23	0.817	0.879
83	8	SVM	93.13	0.979	0.966	95.07	89.47	0.848	0.900
112	7	SVM	93.69	0.982	0.972	96.00	89.34	0.859	0.907
156	6	SVM	94.35	0.984	0.974	95.43	92.31	0.876	0.919
216	5	SVM	95.89	0.992	0.987	97.36	93.12	0.909	0.940
331	4	SVM	96.36	0.994	0.990	97.71	93.79	0.919	0.947
488	3	SVM	96.36	0.995	0.991	98.07	93.12	0.919	0.946
816	2	SVM	96.82	0.996	0.993	98.36	93.93	0.930	0.953
1731	1	SVM	96.92	0.995	0.991	98.50	93.93	0.932	0.955

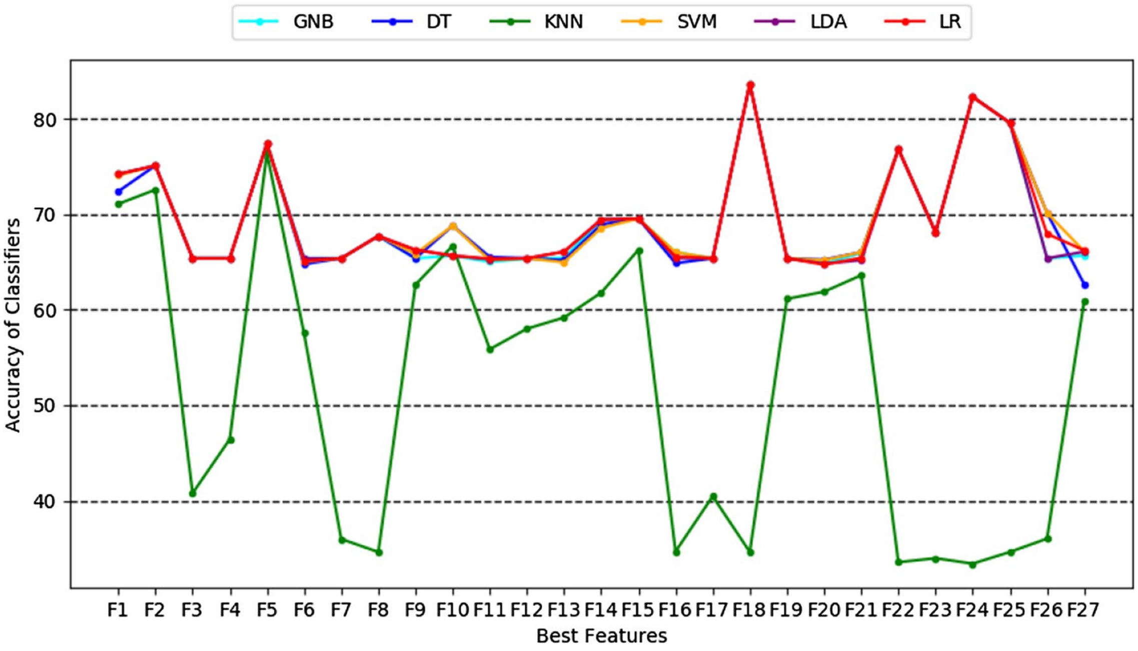

Fig. 8 The accuracy of best 27 individual features by six popular classifiers

Table 8 Performance comparison of iPro70-FMWin with other different method

Method	Acc (%)	auROC	auPR	S_p (%)	S_n (%)	MCC	F_1
IPMD	89.2	0.953	0.920	91.4	**84.9**	0.761	–
iPro70-PseZNC	84.5	0.909	–	86.8	80.3	0.663	–
Z-curve	77.8	0.848	–	79.5	74.6	0.527	–
iPro70-FMWin	**90.57**	**0.959**	**0.937**	**94.43**	83.27	**0.789**	**0.859**

Bold values indicate the best values

our sequence-based predictor named “iPro70-FMWin” for identifying σ^{70} promoter in the prokaryote. For the benefit of researchers, a user-friendly online service was built and can be freely accessible at http://ipro70.pythonanywhere.com/server.

Discussion

In this paper, we have proposed iPro70-FMWin. We have used an optimal number of features extracted from sequences only. We believe that our features are more effective to identify σ^{70} promoter sequences. In addition, we also believe that our 27 features may be useful to identify the other promoter sequences. The future work will focus on different promoter sequences and comparison study among them. Moreover, this role can play a significant supplementary in the other marginalized way for the prediction of promoters and transcription start sites. Besides, there is still a great place to improve the accuracy of prediction. We believe the web application and the method proposed in this paper will play an important role in the field of genomic analysis.

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Aggarwala V, Voight BF (2015) An expanded sequence context model broadly explains variability in polymorphism levels across the human genome. Nat Genet 47(3):349

Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat 46(3):175–185

Arif M, Hayat M, Jan Z (2018) iMem-2LSAAC: a two-level model for discrimination of membrane proteins and their types by extending the notion of SAAC into Chou's pseudo amino acid composition. J Theor Biol 442:11–21

Audic S, Claverie JM (1997) Detection of eukaryotic promoters using Markov transition matrices. Comput Chem 21(4):223–227

Bermingham ML, Pong-Wong R, Spiliopoulou A, Hayward C, Rudan I, Campbell H, Wright AF, Wilson JF, Agakov F, Navarro P (2015) Application of high-dimensional feature selection: evaluation for genomic prediction in man. Sci Rep 5:10312

Boyle AP, Davis S, Shulha HP, Meltzer P, Margulies EH, Weng Z, Furey TS, Crawford GE (2008) High-resolution mapping and characterization of open chromatin across the genome. Cell 132(2):311–322

Chen W, Feng PM, Lin H, Chou KC (2013) iRSpot-PseDNC: identify recombination spots with pseudo dinucleotide composition. Nucleic Acids Res 41(6):e68–e68

Chen W, Lei TY, Jin DC, Lin H, Chou KC (2014) PseKNC: a flexible web server for generating pseudo k-tuple nucleotide composition. Anal Biochem 456:53–60

Chen W, Lin H, Chou KC (2015) Pseudo nucleotide composition or PseKNC: an effective formulation for analyzing genomic sequences. Mol BioSyst 11(10):2620–2634

Chen W, Feng P, Yang H, Ding H, Lin H, Chou KC (2018) iRNA-3typeA: identifying three types of modification at RNAs adenosine sites. Mol Ther Nucleic Acids 11:468–474. https://doi.org/10.1016/j.omtn.2018.03.012

Chen XX, Tang H, Li WC, Wu H, Chen W, Ding H, Lin H (2016) Identification of bacterial cell wall lyases via pseudo amino acid composition. BioMed Res Int 2016:2016

Chou KC (2001a) Prediction of protein cellular attributes using pseudo-amino acid composition. Proteins Struct Funct Bioinf 43(3):246–255

Chou KC (2001b) Prediction of signal peptides using scaled window. Peptides 22(12):1973–1979

Chou KC (2001c) Using subsite coupling to predict signal peptides. Protein Eng 14(2):75–79

Chou KC (2004) Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. Bioinformatics 21(1):10–19

Chou KC (2009) Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology. Curr Proteom 6(4):262–274

Chou KC (2011a) Some remarks on protein attribute prediction and pseudo amino acid composition. J Theor Biol 273(1):236–247

Chou KC (2011b) Some remarks on protein attribute prediction and pseudo amino acid composition. J Theor Biol 273(1):236–247

Chou KC (2013) Some remarks on predicting multi-label attributes in molecular biosystems. Mol BioSyst 9(6):1092–1100

Chou KC (2015) Impacts of bioinformatics to medicinal chemistry. Med Chem 11(3):218–234

Chou KC (2017) An unprecedented revolution in medicinal chemistry driven by the progress of biological science. Curr Top Med Chem 17(21):2337–2358

Chou KC, Shen HB (2009) Recent advances in developing web-servers for predicting protein attributes. Nat Sci 1(02):63

Compeau PE, Pevzner PA, Tesler G (2011) How to apply de Bruijn graphs to genome assembly. Nat Biotechnol 29(11):987

Contreras-Torres E (2018) Predicting structural classes of proteins by incorporating their global and local physicochemical and conformational properties into general Chou's pseaac. J Theor Biol. https://doi.org/10.1016/j.jtbi.2018.05.033

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

Coussement K, Van den Poel D (2008) Churn prediction in subscription services: an application of support vector machines while comparing two parameter-selection techniques. Exp Syst Appl 34(1):313–327

Crawford GE, Holt IE, Whittle J, Webb BD, Tai D, Davis S, Margulies EH, Chen Y, Bernat JA, Ginsburg D (2006) Genome-wide mapping of DNase hypersensitive sites using massively parallel signature sequencing (MPSS). Genome Res 16(1):123–131

Dash M, Liu H (1997) Feature selection for classification. Int Data Anal 1(3):131–156

Demeler B, Zhou G (1991) Neural network optimization for *E. coli* promoter prediction. Nucleic Acids Res 19(7):1593–1599

El Hassan M, Calladine C (1996) Propeller-twisting of base-pairs and the conformational mobility of dinucleotide steps in DNA. J Mol Biol 259(1):95–103

Feng P, Yang H, Ding H, Lin H, Chen W, Chou KC (2018) iDNA6mA-PseKNC: identifying dna n6-methyladenosine sites by incorporating nucleotide physicochemical properties into PseKNC. Genomics. https://doi.org/10.1016/j.ygeno.2018.01.005

Fickett JW, Hatzigeorgiou AG (1997) Eukaryotic promoter recognition. Genome Res 7(9):861–878

Gama-Castro S, Salgado H, Santos-Zavaleta A, Ledezma-Tejeida D, Muñiz-Rascado L, García-Sotelo JS, Alquicira-Hernández K, Martínez-Flores I, Pannier L, Castro-Mondragón JA (2015) RegulonDB version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond. Nucleic Acids Res 44(D1):D133–D143

Gan Y, Guan J, Zhou S (2012) A comparison study on feature selection of DNA structural properties for promoter prediction. BMC Bioinf 13(1):4

Ginno PA, Lim YW, Lott PL, Korf I, Chédin F (2013) Gc skew at the 5' and 3' ends of human genes links r-loop formation to epigenetic regulation and transcription termination. Genome Res 23(10):1590–1600

Gordon JJ, Towsey MW, Hogan JM, Mathews SA, Timms P (2005) Improved prediction of bacterial transcription start sites. Bioinformatics 22(2):142–148

Gordon L, Chervonenkis AY, Gammerman AJ, Shahmuradov IA, Solovyev VV (2003) Sequence alignment kernel for recognition of promoter regions. Bioinformatics 19(15):1964–1971

Grech B, Maetschke S, Mathews S, Timms P (2007) Genome-wide analysis of chlamydiae for promoters that phylogenetically footprint. Res Micro 158(8–9):685–693

Gruber TM, Gross CA (2003) Multiple sigma subunits and the partitioning of bacterial transcription space. Ann Rev Micro 57(1):441–466

Guo SH, Deng EZ, Xu LQ, Ding H, Lin H, Chen W, Chou KC (2014) iNuc-PseKNC: a sequence-based predictor for predicting nucleosome positioning in genomes with pseudo k-tuple nucleotide composition. Bioinformatics 30(11):1522–1529

Hanley JA, McNeil BJ (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143(1):29–36

Hodge V, Austin J (2004) A survey of outlier detection methodologies. Artif Int Rev 22(2):85–126

Hosmer DW Jr, Lemeshow S, Sturdivant RX (2013) Applied logistic regression, vol 398. Wiley, Oxford

Huerta AM, Collado-Vides J (2003) Sigma70 promoters in *Escherichia coli*: specific transcription in dense regions of overlapping promoter-like signals. J Mol Biol 333(2):261–278

James G, Witten D, Hastie T, Tibshirani R (2013) An introduction to statistical learning, vol 112. Springer, Berlin

Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. IJCAI Montreal Can 14:1137–1145

Krishnan SM (2018) Using Chou's general PseAAC to analyze the evolutionary relationship of receptor associated proteins (RAP) with various folding patterns of protein domains. J Theor Biol 445:62–74

Li FM, Li QZ (2008) Using pseudo amino acid composition to predict protein subnuclear location with improved hybrid approach. Amino Acids 34(1):119–125

Li QZ, Lin H (2006) The recognition and prediction of σ70 promoters in *Escherichia coli* k-12. J Theor Biol 242(1):135–141

Liang ZY, Lai HY, Yang H, Zhang CJ, Yang H, Wei HH, Chen XX, Zhao YW, Su ZD, Li WC et al (2017) Pro54db: a database for experimentally verified sigma-54 promoters. Bioinformatics 33(3):467–469

Lin H, Li QZ (2011) Eukaryotic and prokaryotic promoter prediction using hybrid approach. Theory Biosci 130(2):91–100

Lin H, Deng EZ, Ding H, Chen W, Chou KC (2014) iPro54-PseKNC: a sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition. Nucleic Acids Res 42(21):12961–12972

Lin H, Liang Z, Tang H, Chen W (2017) Identifying sigma70 promoters with novel pseudo nucleotide composition. IEEE ACM Trans Comput Biol Bioinf 2017:10

Liu B, Liu F, Wang X, Chen J, Fang L, Chou KC (2015) Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. Nucleic Acids Res 43(W1):W65–W71

Liu B, Wu H, Chou KC (2017a) Pse-in-One 2.0: an improved package of web servers for generating various modes of pseudo components of DNA, RNA, and protein sequences. Nat Sci 9(04):67

Liu B, Yang F, Huang DS, Chou KC (2017b) iPromoter-2L: a two-layer predictor for identifying promoters and their types by multi-window-based PseKNC. Bioinformatics 34(1):33–40

Liu B, Li K, Huang DS, Chou KC (2018a) iEnhancer-EL: identifying enhancers and their strength with ensemble learning approach. Bioinformatics. https://doi.org/10.1093/bioinformatics/bty458

Liu B, Weng F, Huang DS, Chou KC (2018b) iRO-3wPseKNC: Identify DNA replication origins by three-window-based PseKNC. Bioinformatics 1:8

Lobry J (1996) Asymmetric substitution patterns in the two DNA strands of bacteria. Mol Biol Evol 13(5):660–665

Lukashin A, Anshelevich V, Amirikyan B, Gragerov A, Frank-Kamenetskii M (1989) Neural network models for promoter recognition. J Biomol Struct Dyn 6(6):1123–1133

Mallios RR, Ojcius DM, Ardell DH (2009) An iterative strategy combining biophysical criteria and duration hidden Markov models for structural predictions of *Chlamydia trachomatis*σ 66 promoters. BMC Bioinf 10(1):271

Mei J, Zhao J (2018a) Analysis and prediction of presynaptic and postsynaptic neurotoxins by Chou's general pseudo amino acid composition and motif features. J Theor Biol 447:147–153

Mei J, Zhao J (2018b) Prediction of HIV-1 and HIV-2 proteins by using Chous pseudo amino acid compositions and different classifiers. Sci Rep 8(1):2359

Mika S, Ratsch G, Weston J, Scholkopf B, Mullers KR (1999) Fisher discriminant analysis with kernels. In: Neural networks for signal processing IX, 1999. Proceedings of the 1999 IEEE signal processing society workshop, IEEE, pp 41–48

Murphy KP (2006) Naive Bayes classifiers. University of British Columbia, Vancouver, p 18

Olson DG, Maloney M, Lanahan AA, Hon S, Hauser LJ, Lynd LR (2015) Identifying promoters for gene expression in *Clostridium thermocellum*. Metab Eng Commun 2:23–29

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12(Oct):2825–2830

Rahman MS, Shatabda S, Saha S, Kaykobad M, Rahman MS (2018a) DPP-PseAAC: a DNA-binding protein prediction model using Chous general PseAAC. J Theor Biol 452:22–34

Rahman S, Aktar U, Jani R, Shatabda S (2018b) iPromoter-FSEn: identification of bacterial σ70 promoter sequences using feature subspace based ensemble classifier. Genomics. https://doi.org/10.1016/j.ygeno.2018.07.011

Rayhan F, Ahmed S, Shatabda S, Farid DM, Mousavian Z, Dehzangi A, Rahman MS (2017) idti-esboost: identification of drug target interaction using evolutionary and structural features with boosting. Sci Rep 7(1):17731

Sabooh MF, Iqbal N, Khan M, Khan M, Maqbool H (2018) Identifying 5-methylcytosine sites in rna sequence using composite encoding feature into Chou's PseKNC. J Theor Biol 452:1–9

Safavian SR, Landgrebe D (1991) A survey of decision tree classifier methodology. IEEE Trans Syst Man Cybern 21(3):660–674

Samocha KE, Robinson EB, Sanders SJ, Stevens C, Sabo A, McGrath LM, Kosmicki JA, Rehnström K, Mallick S, Kirby A (2014) A framework for the interpretation of de novo mutation in human disease. Nat Genet 46(9):944

Shen L, Bai L (2004) AdaBoost Gabor feature selection for classification. In: Proceedings of image and vision computing, New Zealand, pp 77–83

Shin J, Noireaux V (2010) Efficient cell-free expression with the endogenous *E. coli* RNA polymerase and sigma factor 70. J Biol Eng 4(1):8

e Silva SDA, Forte F, Sartor IT, Andrighetti T, Gerhardt GJ, Delamare APL, Echeverrigaray S (2014) Dna duplex stability as discriminative characteristic for *Escherichia coli*σ54-and σ28-dependent promoter sequences. Biologicals 42(1):22–28

Song K (2011) Recognition of prokaryotic promoters based on a novel variable-window z-curve method. Nucleic Acids Res 40(3):963–971

Stormo GD (2000) Dna binding sites: representation and discovery. Bioinformatics 16(1):16–23

Su ZD, Huang Y, Zhang ZY, Zhao YW, Wang D, Chen W, Chou KC, Lin H (2018) iLoc-lncRNA: predict the subcellular location of lncrnas by incorporating octamer composition into general PseKNC. Bioinformatics. https://doi.org/10.1093/bioinformatics/bty508

Tang H, Chen W, Lin H (2016) Identification of immunoglobulins using Chou's pseudo amino acid composition with feature selection technique. Mol BioSyst 12(4):1269–1275

Tang H, Zhao YW, Zou P, Zhang CM, Chen R, Huang P, Lin H (2018) Hbpred: a tool to identify growth hormone-binding proteins. Int J Biol Sci 14(8):957–964

Towsey M, Timms P, Hogan J, Mathews SA (2008) The cross-species prediction of bacterial promoters using a support vector machine. Comput Biol Chem 32(5):359–366

Williamson DF, Parker RA, Kendrick JS (1989) The box plot: a simple visual method to interpret data. Ann Intern Med 110(11):916–921

Yamagishi H (1974) Nucleotide distribution in bacterial DNA's differing in g+ c content. J Mol Evol 3(3):239–242

Yang H, Tang H, Chen XX, Zhang CJ, Zhu PP, Ding H, Chen W, Lin H (2016) Identification of secretory proteins in mycobacterium tuberculosis using pseudo amino acid composition. BioMed Res Int

Yang H, Qiu WR, Liu G, Guo FB, Chen W, Chou KC, Lin H (2018) iRSpot-Pse6NC: identifying recombination spots in *Saccharomyces cerevisiae* by incorporating hexamer composition into general PseKNC. Int J Biol Sci 14(8):883

Zhang CJ, Tang H, Li WC, Lin H, Chen W, Chou KC (2016) iOri-human: identify human origin of replication by incorporating dinucleotide physicochemical properties into pseudo nucleotide composition. Oncotarget 7(43):69783

Zhang S, Duan X (2018) Prediction of protein subcellular localization with oversampling approach and Chou's general PseAAC. J Theor Biol 437:239–250