scientific reports

Check for updates

OPEN

A machine learning technique for identifying DNA enhancer regions utilizing CIS-regulatory element patterns

Ahmad Hassan Butt[1], Tamim Alkhalifah[2]✉, Fahad Alturise[2] & Yaser Daanial Khan[1]

Enhancers regulate gene expression, by playing a crucial role in the synthesis of RNAs and proteins. They do not directly encode proteins or RNA molecules. In order to control gene expression, it is important to predict enhancers and their potency. Given their distance from the target gene, lack of common motifs, and tissue/cell specificity, enhancer regions are thought to be difficult to predict in DNA sequences. Recently, a number of bioinformatics tools were created to distinguish enhancers from other regulatory components and to pinpoint their advantages. However, because the quality of its prediction method needs to be improved, its practical application value must also be improved. Based on nucleotide composition and statistical moment-based features, the current study suggests a novel method for identifying enhancers and non-enhancers and evaluating their strength. The proposed study outperformed state-of-the-art techniques using fivefold and tenfold cross-validation in terms of accuracy. The accuracy from the current study results in 86.5% and 72.3% in enhancer site and its strength prediction respectively. The results of the suggested methodology point to the potential for more efficient and successful outcomes when statistical moment-based features are used. The current study's source code is available to the research community at https://github.com/csbioinfopk/enpred.

In cellular biology, regulation of transcription is performed to recruit elongation factors or RNA polymerase II initiation. This is mainly achieved at specific sequences of DNA by binding transcriptional factors (TFs). Transcription initiation sites are harbored by promoter regions which are the most studied sites in DNA[1]. Some DNA sequences have multiple transcription factor binding sites and are near or far away from promoter regions. Such DNA segments are denoted as enhancers[2,3].The transcription of genes is boosted by enhancers which influence various cellular activities such as cell carcinogenesis and virus activity, tissue specificity of gene expression, differentiation and cell growth, regulation and gene expression and develop relationship between such processes very closely[4].

Enhancers can be a short (50–1500 bp) segment of DNA and situated 1Mbp (1,000,000 bp) distance away from a gene. Sometimes they can even exist in different chromosomes[5,6]. On the other hand, promoters are located near the start of the transcription sites of a gene. Due to this fact of locational difference between promoters and enhancers, the task of enhancer's prediction is highly difficult and challenging than promoters[7]. Many human diseases like inflammatory bowel disease, disorder and various cancers have been linked to this genetic variation in enhancers[8–11].

A DNA segment characterized as the first enhancer, reported 40 years ago, increased the transcription of β-globin gene during a transgenic assay inside the virus genome of SV40 tumor[12]. Scientific research during recent past has discovered that enhancers have many subgroups such as weak and strong enhancers, latent enhancers and poised enhancers[13]. Prediction of enhancers and their subgroups is an interesting area of research as they are considered important in disease and evolution. In higher classification of eukaryotes, transcription factor repertoire, diverse in nature, binds to enhancers[14]. This process of binding orchestrates many cellular events that are critical to the cellular system. Some of the events that are coordinated through this binding are maintenance of the cell identity, differentiation and response to stimuli[15,16].

[1]Department of Computer Science, School of Systems and Technology, University of Management and Technology, Lahore, Pakistan. [2]Department of Computer, College of Science and Arts in Ar Rass, Qassim University, Ar Rass, Saudi Arabia. ✉email: tkhliefh@qu.edu.sa

In the past, purely experimental techniques were being relied upon for the prediction of enhancers. Pioneering works in enhancer prediction was proposed in[4,17]. The former was to use combinations such as transcription factor, P300[18], with enhancers to identify them. This method would usually under-detect or miss the concerned targets. This has resulted in high failure rates because all enhancers do not have transcription factor occupations. The latter was to utilize DNase I hypersensitivity for enhancer predictions. Hence, this led to a high false-positive rate as many other DNA segments, which were non-enhancers, were detected incorrectly as enhancers. Although, genome-wide mapping techniques of histone modifications[1,19–23] could improve the aforesaid deficiencies in the prediction of promoters and enhancers, but they are time consuming and expensive.

Several bioinformatics tools have been developed for rapid and cost effective classification of enhancers in genomics. CSI-ANN[21] used data transformations efficiently to formulate the samples and predict using Artificial-Neural-Network (ANN) classifications. EnhancerFinder[1] incorporated evolutionary conservation information features into sample formulation combined with a multiple kernel learning algorithm as a classifier. RFECS[23]applied random forest algorithm for improvements in detection methods. EnhancerDBN[24] used deep belief networks for enhancer predictions. BiRen[25] increased the predictive performance by utilizing deep learning based method. By utilizing these bioinformatics tools, enhancer detection can be achieved by the research community. Formed using many different large sub-groups of functional elements, enhancers can be grouped as weak, strong, inactive and poised enhancers. iEnhancer-2L[26],the first ever predictor to detect enhancers and identify their strengths and was based on sequence information only. Pseudo K-tuple nucleotide compositions (PseKNC) based features were incorporated into iEnhancer-2L. It has been used in many analysis related to genomics increasingly. Furthermore, many other methods, such as EnhancerPred[27] and EnhancerPred_2.0[28], were introduced to improve the performance by incorporating other features based on DNA sequences. iEnhancer-5Step[29] was recently developed using the hidden information of DNA sequences infused with Support Vector Machine (SVM) based predictions. Recently, iEnhancer-RD[30] combined features and utilized recursive feature elimination algorithm for feature selection with deep neural network for enhancer identification. Similarly, ES-ARCNN[31] used reverse complement method of data augmentation with residual Convolution Neural Network (CNN) to predict enhancer strength. iEnhancer-GAN[32] also implemented CNNs to identify enhancers with strength using deep learning frameworks and combination of word embedding techniques. iEnhancer-XG[33] utilized XGBoost classifier as base classifier and five feature extraction methods namely, K-Spectrum Profile, Mismatch K-tuple, Subsequence Profile, Position-Specific-Scoring-Matrix (PSSM) and Pseudo dinucleotide composition (PseDNC) to classify enhancers and their strength. iEnhancer-KL[34] also implemented Position specific Nucleotide Composition and Kullback–Leibler (KL) method with several machine learning models. Enhancer-IF utilized comprehensively explored heterogeneous features with five commonly used machine learning algorithms. These five methods were extensively trained using 35 baseline models having seven encodings. This integration of five meta–models enhanced the overall performance of prediction model. BERT(bidirectional encoder representations from transformers)[35] and 2D CNN based models were used with the contextualized word embedding for capturing the semantics and context of the words for representing DNA sequences. This opened a new avenue in biological sequence modeling. iEnhancer-MFGBDT[36] used gradient boosting decision tree by fusing multiple features which included k-mer, k-mer with reverse compliments, second-order moving components etc. compared to other state of the art methods, this was an effective and intelligent tool to identify enhancers. iEnhancer-ECNN[37] used one hot encoding methods and k-mers for data transformation and convolution neural networks (CNN) for identifying enhancers and classify their strengths. An ensemble deep recurrent neural network based method[38] was also used to identify enhancers and their strength. These deep ensemble networks were generated from six types of dinucleotide physiochemical properties. These properties outperformed other features and achieved better performance and efficiency. This method proved to be better and has the potential to improve performance of biological sequential modeling using shallow machine learning models. However, improvement in the performance of the aforementioned predictors is still required. Specifically, the success rate of discriminating strong and weak enhancers is not up to the expectations of the scientific community. The current study is initiated to propose a method which would deal with this problem.

Materials and methods

Benchmark dataset. The benchmark dataset of DNA enhancer sites, originally constructed and used in recent past by iEnhancer-2L[26], was re-used in the proposed method. In the current dataset, information related to nine different cell lines (K562, H1ES, HepG2, GM12878, HSMM, HUVEC, NHEK, NHLF and HMEC) was used in the collection of enhancers and 200 bp fragments were extracted from DNA sequences. The annotation of chromatin state information was performed by ChromHMM. The whole genome profile included multiple histone marks such as, H3K27ac H3K4me1, H3K4me3, etc. To remove pairwise sequences from the dataset, CD-HIT[39] tool was used to remove sequences having more than 20% similarity. The benchmark dataset includes 2968 DNA enhancer sequences from which 1484 are non-enhancer sequences and 1484 are enhancer sequences. From 1484 enhancer sequences, 742 are strong enhancers and 742 are weak enhancers for the second layer classification. Furthermore, the independent dataset used by iEnhancer-5Step[29] was utilized to enhance the effectiveness and performance of the proposed model. The independent dataset included 400DNA enhancer sequences from which 200 (100 strong and 100 weak enhancers) are enhancers and 200 are non-enhancers. Table 1 includes the breakdown of the benchmark dataset. The details of the above mention dataset is provided in the Supplementary Material (see Online Supporting Information S1, Online Supporting Information S2 and Online Supporting Information S3).

It is not always simple to understand the semantics of a piece of data, which in turn reflects the difficulty of developing biological data models. It can be difficult to come to a consensus about the data in a given domain because different people will emphasize different features, use different terminology, and have different

DNA samples	Benchmark dataset[26]	Independent dataset[29]
Non-enhancers	1484	200
Enhancers	1484	200
Overall	2968	400
Breakdown of strong and weak enhancers dataset		
Strong enhancers	742	100
Weak enhancers	742	100
Total enhancer	1484	200

Table 1. Breakdown of the benchmark datasets of DNA enhancers and non-enhancers.

perspectives on how things should be seen. The fact that biosciences are non-axiomatic and that different, though closely related communities have very different perspectives on the same or similar concepts makes the situation even more difficult. Biological data models, however, can be useful for creating, making explicit, and communicating precise and in-depth descriptions of data that is already available or soon to be produced. It is hoped that the current study will increase the use of biological data models in bioinformatics, alleviating the management and sharing issues that are currently becoming more and more problematic.

In statistical based prediction models, the benchmark dataset mostly includes training datasets and testing datasets. By utilizing various benchmark datasets, results obtained are computed from fivefold and tenfold cross-validations. The definition of a benchmark dataset is used in Eq. (1):

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

where D^{+} contains 1484 enhancers and D^{-} contains 1484 non-enhancers. D^{+}_{strong} contains 742 strong enhancers, D^{+}_{weak} contains 742 weak enhancers and U denotes the symbol of "union" in the set theory.

Feature extraction. An effective bioinformatics predictor is the need of researchers in medicine and pharmacology to formulate the biological sequence with a vector or a discrete model without losing any key-order characteristics or sequence-pattern information. The reason for this fact, as explained in a comprehensive state-of-the-art review[40], that the existing machine-learning algorithms cannot handle sequences directly but rather in vector formulations. However, there exists some possibility that all the sequence-pattern information from a vector might be lost in a discrete model formulation. To overcome the sequence-pattern information loss from proteins, Chou proposed pseudo amino acid composition (PseAAC)[41]. In almost all areas of bioinformatics and computational proteomics[40], the Chou's PseAAC concept has been widely used ever since it was proposed. In the recent past, three publicly accessible and powerful softwares, 'propy'[42], 'PseAAC-Builder'[43] and 'PseAAC-General'[44] were developed and the importance and popularity of Chou's PseAAC in computational proteomics has increased more ever since. 'PseAAC-General' calculates Chou's general PseAAC[45] and the other two software generate Chou's special PseAAC in various modes[46]. The Chou's general PseAAC included not only the feature vectors of all the special modes, but also the feature vectors of higher levels, such as "Gene Ontology" mode[45], "Functional Domain" mode[45] and "Sequential Evolution" mode or "PSSM" mode[45]. Using PseAAC successfully for finding solutions to various problems relevant to peptide/protein sequences, encouraged the idea to introduce PseKNC (Pseudo K-tuple Nucleotide Composition)[47] for generating different feature vectors for DNA/RNA sequences[48,49] which proved very effective and efficient as well. In recent times a useful, efficient and a very powerful webserver called 'Pse-in-One'[50] and its recently updated version 'Pse-in-One2.0'[51] were developed that are able to generate any preferred feature vector of pseudo components for DNA/RNA and protein/peptide sequences.

In this study, we utilized the Kmer[52] approach to represent the DNA sequences. According to Kmer, the occurrence frequency of 'n' neighboring nucleic acids can be represented from a DNA sequence. Hence, by using the sequential model, a sample of DNA having 'w' nucleotides is expressed generally as Eq. (2)

$$\mathbf{S} = Y_1 Y_2 Y_3 \ldots Y_v \ldots Y_w \tag{2}$$

where Y_1 is represented as the first nucleotide of the DNA sample **S,** Y_2 as the second nucleotide having the 2nd position of occurrence in DNA sample S and so on so fourth Y_w denotes the last nucleotide of the DNA sample. 'w' is the total length of the nucleotides in a DNA sample. The Y_v nucleotide can be any four of the nucleotides which can be represented using the aforementioned discrete model. The nucleotide Y_v can be further described using Eq. (3)

$$Y_v \in \{A(adenine)\ C(cytosine)\ G(guanine)\ T(thymine)\} \tag{3}$$

Here $\in$ is the symbol used to represent the set theory 'member of' property and $1 \le v \le n$. The components that are defined by the aforementioned discrete model utilize relevant nucleotides useful features to expedite the extraction methods. These components are further used in statistical moments based feature extraction methods.

Statistical moments. Statistical moments are quantitative measures that are used for the study of the concentrations of some key configurations in a collection of data used for pattern recognition related problems[53]. Several properties of data are described by different orders of moments. Some moments are used to reveal eccentricity and orientation of data while some are used to estimate the data size[54–59]. Several moments have been formed by various mathematicians and statisticians based on famous distribution functions and polynomials[60–62]. These moments were utilized to explicate the current problem[63].

The moments that are used in calculations of mean, variance and asymmetry of the probability distribution are known as raw moments. They are neither location-invariant nor scale-invariant. Similar type of information is obtained from the Central moments, but these central moments are calculated using the centroid of the data. The central moments are location-invariant with respect to centroid as they are calculated along the centroid of the data, but still they remain scale-variant. The moments based on Hahn polynomials are known as Hahn moments. These moments are neither location-variant nor scale-invariant[64–67]. The fact that these moments are sensitive to biological sequence ordered information amplifies the reason to choose them as they are primarily significant in extracting the obscure features from DNA sequences. These features have been utilized in previous research studies[54,59–61,68–73] and have proved to be more robust and effective in extracting core sequence characteristics. The use of scale-invariant moment has consequently been avoided during the current study. The values quantified from utilizing each method enumerate data on its own measures. Furthermore, the variations in data source characteristics imply variations in the quantified value of moments calculated for arbitrary datasets. In the current study, the 2D version of the aforementioned moments is used and hence the linear structured DNA sequence as expressed by Eq. (2) is transformed into a 2D notation. The DNA sequence, which is 1D, is transformed to a 2D structure using row major scheme through the following Eq. (4):

$$d = \lceil \sqrt{z} \rceil \tag{4}$$

where the sample sequence length is 'z' and the2-dimensional square matrix has '*d*' as its dimension. The ordering obtained from Eq. (4) is used to form matrix M (Eq. 5) having 'm' rows and 'm' columns.

$$M = \begin{bmatrix} N_{1\to1} & N_{1\to2} & \cdots & N_{1\to j} & \cdots & N_{1\to m} \\ N_{2\to1} & N_{2\to2} & \cdots & N_{2\to j} & \cdots & N_{2\to m} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ N_{k\to1} & N_{k\to2} & \cdots & N_{k\to j} & \cdots & N_{k\to m} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ N_{m\to1} & N_{m\to2} & \cdots & N_{m\to j} & \cdots & N_{m\to m} \end{bmatrix} \tag{5}$$

The transformation from M matrix to square matrix M' is performed using the mapping function 'ℝ'. This function is defined as Eq. (6):

$$\mathbb{R}(p_x) = M_{ij} \tag{6}$$

If the population of square matrix M' is done as row major order then, $i = \frac{x}{m} + 1$ and $j = xmodm$.

Any vector or matrix, which represents any pattern, can be used to compute different forms of moments. The values of M' are used to compute raw moments. The moments of a 2D continuous function $A(j,k)$ to order (j + k) are calculated from Eq. (7):

$$A_{jk} = \sum_{a}\sum_{b} a^j b^k f(a,b) \tag{7}$$

The raw moments of 2D matrix M, with order (j + k) and up to a degree of 3,are computed using the Eq. (7). The origin of data as the reference point and distant components from the origin are assumed and utilized by the raw moments for computations. The 10 moment features computed up to degree-3 are labeled as M_{00},M_{01}, M_{10}, M_{11},M_{02}, M_{20}, M_{12},M_{21}, M_{30} and M_{03}.

The centroid of any data is considered as its center of gravity. The centroid is the point in the data where it is uniformly distributed in all directions in the relations of its weighted average[74,75]. The central moments are also computed up to degree-3, using the centroid of the data as their reference point, from the following Eq. (8):

$$\mu_{jk} = \sum_{a}\sum_{b} (a-\overline{a})^j \left(b-\overline{b}\right)^k f(a,b) \tag{8}$$

The degree-3 central moments with ten distinct feature sare labeled as μ_{00}, μ_{01}, μ_{10}, μ_{11},μ_{02},μ_{20},μ_{12}, μ_{21}, μ_{30} &μ_{03}. The centroids $\overline{a}$ and $\overline{b}$ are calculated from Eqs. (9) and (10):

$$\overline{a} = \frac{M_{10}}{M_{00}}, \tag{9}$$

$$\overline{b} = \frac{M_{01}}{M_{00}} \tag{10}$$

The Hahn moments are computed by transforming 1D notations into square matrix notations. This square matrix is valuable for the computations of discrete Hahn moments or orthogonal moments as these moments

are of 2D form and require a two-dimensional square matrix as input data. These Hahn moments are orthogonal in nature that implies that they possess reversible properties. Usage of this property enables the reconstruction of the original data using the inverse functions of discrete Hahn moments. This further indicates that the compositional and positional features of a DNA sequence are somehow conserved within the calculated moments. M' matrix is used as 2D input data for the computations of Orthogonal Hahn moments. The order 'm' Hahn polynomial can be computed from Eq. (11):

$$h_m^{x,y}(i,N) = (N+y-1)_m (N-1)_m \sum_{j=0}^{m} (-1)^j \frac{(-m)_j (-i)_j (2N+x+y-m-1)_j}{(N+y-1)_j (N-1)_j} \cdot \frac{1}{j!} \tag{11}$$

The aforementioned Pochhammer symbol (($\mathfrak{p}$)) was defined as follows in Eq. (12):

$$(\mathfrak{p})_k = \mathfrak{p}(\mathfrak{p}+1)\ldots(\mathfrak{p}+k-1) \tag{12}$$

And was simplified further by the Gamma operator in Eq. (13):

$$(\mathfrak{p})_k = \frac{\Gamma(\mathfrak{p}+k)}{\Gamma(\mathfrak{p})} \tag{13}$$

The Hahn moments raw values are scaled using a weighting function and a square norm given as in Eq. (14):

$$\widetilde{h_m^{x,y}}(i,N) = h_m^{x,y}(i,N)\sqrt{\frac{\rho(i)}{k_m^2}}, m = 0,1,\ldots,N-1 \tag{14}$$

Meanwhile, in Eq. (15),

$$\rho(i) = \frac{\Gamma(x+i+y)\Gamma(y+i+1)(x+y+i+1)_N}{(x+y+2i+1)m!(N-i-1)!} \tag{15}$$

The Hahn moments are computed up to degree-3for the 2-D discrete data as follows in Eq. (16):

$$H_{uv} = \sum_{b=0}^{N-1}\sum_{a=0}^{N-1} \beta_{ab} \widetilde{h_u^{x,y}}(b,N) \widetilde{h_v^{x,y}}(a,N), m,n = 0,1,\ldots,N-1 \tag{16}$$

The 10 key Hahn moments-based features are represented by H_{00}, H_{01}, H_{10}, H_{11},H_{02},H_{20},H_{12},H_{21},H_{30}andH_{03}. Matrix M' was utilized in computing ten Raw, ten Central and ten Hahn moments for every DNA sample sequence up to degree-3 which later are unified into the miscellany super feature vector (SFV).

DNA-position-relative-incident-matrix (D-PRIM). The DNA characteristics such as ordered location of the nucleotides in the DNA sequences are of pivotal significance for identification. The relative positioning of nucleotides in any DNA sequence is considered core patterns prevailing the physical features of the DNA sequence. The DNA sequence is represented by D-PRIM in (4×4) order. The matrix in Eq. (17) is used to extract position-relative attributes of every nucleotide in the given DNA sequence.

$$SD-PRIM = \begin{bmatrix} N_{1\to1} & N_{2\to1} & N_{3\to1} & N_{4\to1} \\ N_{1\to2} & N_{2\to3} & N_{3\to2} & N_{4\to2} \\ N_{1\to3} & N_{2\to3} & N_{3\to3} & N_{4\to3} \\ N_{1\to4} & N_{2\to4} & N_{3\to4} & N_{4\to4} \end{bmatrix} \tag{17}$$

The position occurrence values of nucleotides are represented here using the notation $N_{x\to y}$. Here the indication score of the yth position nucleotide is determined using $N_{x\to y}$ with respect to the xth nucleotide first occurrence in the sequence. The nucleotide type 'y'substitutes this score in the biological evolutionary process. The occurrence positional values, in alphabetical order, represented as four native nucleotides. The $S_{D\text{-}PRIM}$ matrix is formed with 16 coefficient values obtained after successfully performing computations on position relative incidences. Similarly, $S_{D\text{-}PRIM16}$[68]and $S_{D\text{-}PRIM64}$[68]were constructedhaving16×16 and 64×64 valuable coefficient features respectively. The 2D heatmaps of these matrices are shown in Figs. 1, 2 and 3. These heatmaps are based on the summation of nucleotide, dinucleotide and trinucleotide composition PRIMs.

30 raw, central and Hahn moments (10 raw, 10 central & 10 Hahn), up to degree-3, were computed using the 2D $S_{D\text{-}PRIM}$ matrix through which 30 features were obtained with 16 unique coefficients and were further incorporated into the miscellany Super Feature Vector (SFV).

DNA-reverse-position-relative-incident-matrix (D-RPRIM). It often happens in cellular biology that the same ancestor is responsible for evolving more than one DNA sequence. These cases mostly outcome homologous sequences. The performance of the classifier is hugely affected by these homologous sequences and hence for producing accurate results, sequence similarity searching is reliable and effectively useful. In machine learning, accuracy and efficiency is hugely dependent on the meticulousness and thoroughness of algorithms through which most pertinent features in the data are extracted. The algorithms used in machine learning have the ability to learn and adapt the most obscure patterns embedded in the data while understanding and uncovering them during the learning phase. The procedure followed during the computation of D-PRIM was utilized in computations of D-RPRIM but only with reverse DNA sequence ordering. The position occurrence values of nucleotides

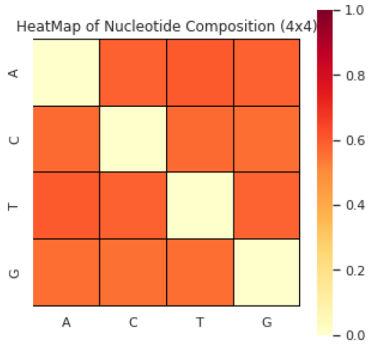

Figure 1. The heatmap of nucleotide composition based PRIMs.

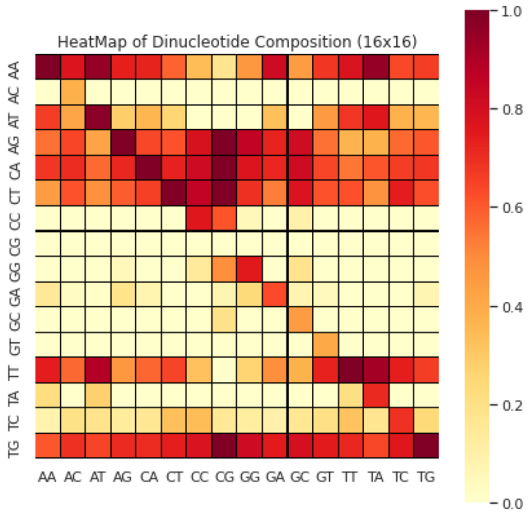

Figure 2. The heatmap of dinucleotide composition based PRIMs.

are represented here using the notation $N_{x\to y}$. Here the indication score of the y [th]position nucleotide is determined using $N_{x\to y}$ with respect to the xth nucleotide first occurrence in the sequence. The nucleotide type 'y' substitutes this score in the biological evolutionary process. The occurrence positional values, in alphabetical order, represented as 4 native nucleotides. This procedure further uncovered hidden patterns for prediction and ambiguities between similar DNA sequences were also alleviated. The 2D matrix D-RPRIM was formed with (4×4) order having16unique coefficients. It is defined by Eq. (18):

$$SD-RPRIM=\begin{bmatrix} N_{1\to1} & N_{2\to1} & N_{3\to1} & N_{4\to1} \\ N_{1\to2} & N_{2\to3} & N_{3\to2} & N_{4\to2} \\ N_{1\to3} & N_{2\to3} & N_{3\to3} & N_{4\to3} \\ N_{1\to4} & N_{2\to4} & N_{3\to4} & N_{4\to4} \end{bmatrix} \tag{18}$$

Similarly, 30 raw, central and Hahn moments (10 raw, 10 central & 10 Hahn), up to degree-3, were computed using the 2D $S_{D\text{-}RPRIM}$ matrix through which 30 features were also obtained with 16 unique coefficients and they were also incorporated into the miscellany Super Feature Vector (SFV).

Frequency-distribution-vector (FDV). The distribution of occurrence of every nucleotide was used to compute the frequency distribution vector. The frequency distribution vector (FDV) is defined as in Eq. (19):

$$\alpha=\{\mathfrak{p}_i,\ldots,\mathfrak{p}_4\} \tag{19}$$

Here ρ_i is the frequency of occurrence of the ith $(1\le i\le4)$ relevant nucleotide. Furthermore, the relative positions of nucleotides in any sequence are highly utilized using these measures. The miscellany Super Feature Vector (SFV) includes these four features from FDV as unique attributes. The violin plots of nucleotide composition and overall frequency normalization is shown in Figs. 4a–d and 5.

D-AAPIV (DNA-accumulative-absolute-position-incidence-vector). The distributional information of nucleotides was stored using frequency distribution vector which used the hidden patterns features of DNA sequences

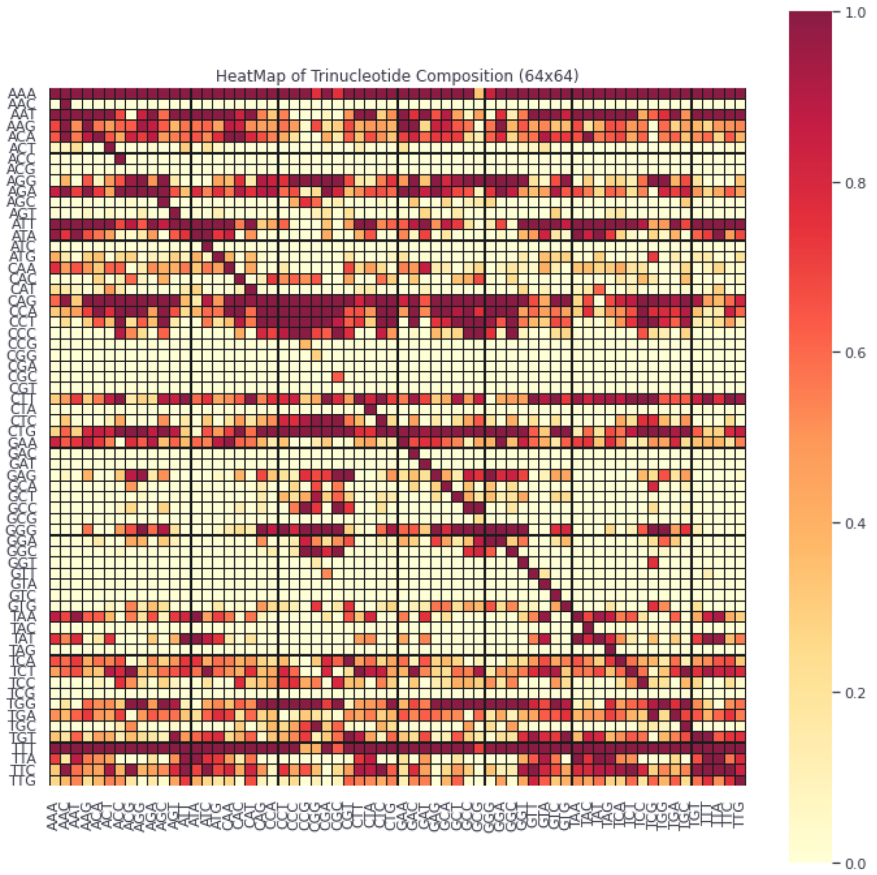

Figure 3. The heatmap of trinucleotide composition based PRIMs.

in relevance to the compositional details. FDV does not have any information regarding relative positional details of relevant nucleotide residues in DNA sequences. This relative positional information was accommodated using D-AAPIV with a length of four critical features associated with four native nucleotides in a DNA sequence. These four critical features from D-AAPIV are also added into the miscellany Super Feature Vector (SFV).

$$\text{D-}AAPIV = \{\text{ɔ}_i, \ldots, \text{ɔ}_4\} \tag{20}$$

Here α_i is any element of D-AAPIV, from DNA sequence S_j having 'n' total nucleotides, which can be calculated using Eq. (21):

$$\beta_i = \sum_{j=1}^{n} S_j \tag{21}$$

D-RAAPIV (DNA-reverse-accumulative-absolute-position-incidence-vector). D-RAAPIV is calculated using the reverse DNA sequence as input with the same method used using D-AAPIV calculations. This vector is calculated to find the deep and hidden features of every sample with respect to reverse relative positional information. D-RAAPIV is formed as the following Eq. (24) using the reversed DNA sequence and generates four valuable features. These four critical features from D-RAAPIV are also added into the miscellany Super Feature Vector (SFV).

$$\text{D-R}AAPIV = \{\text{ɔ}_i, \ldots, \text{ɔ}_4\} \tag{22}$$

Here α_i is any element of D-RAAPIV, from DNA sequence S_j having 'n' total nucleotides, which can be calculated using Eq. (23):

$$\beta_i = \sum_{j=1}^{n} Reverse(S)_j \tag{23}$$

After calculating all possible features from the aforementioned extraction methods, the Super Feature Vector (SFV)was constructed, for further processing in classification algorithm. The proposed model has used extracted

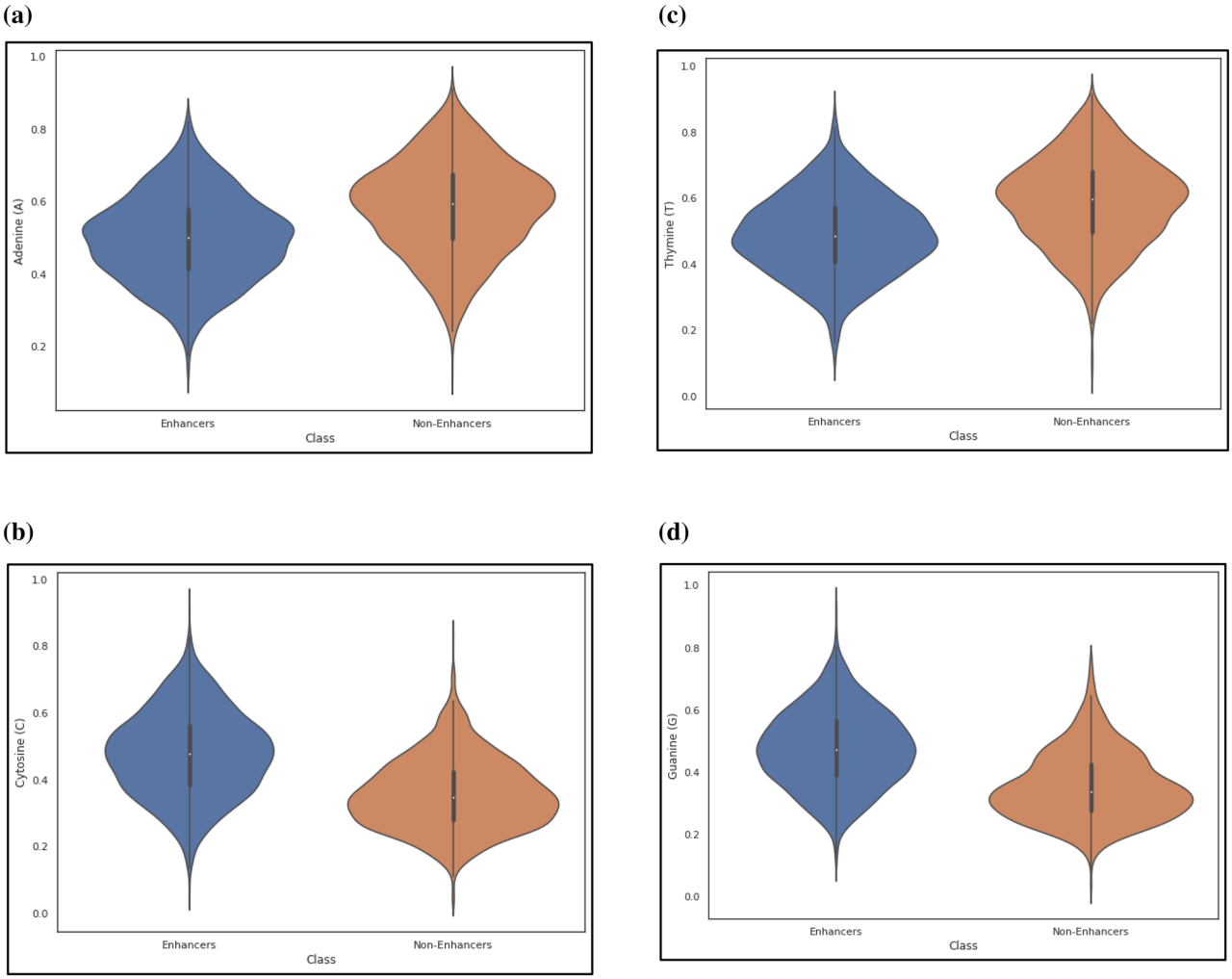

Figure 4. (**a**) The violin plot of nucleotide adenine (A) composition. (**b**) The violin plot of nucleotide cytosine (C) composition. (**c**) The violin plot of nucleotide thymine (T) composition. (**d**) The violin plot of nucleotide guanine (G) composition.

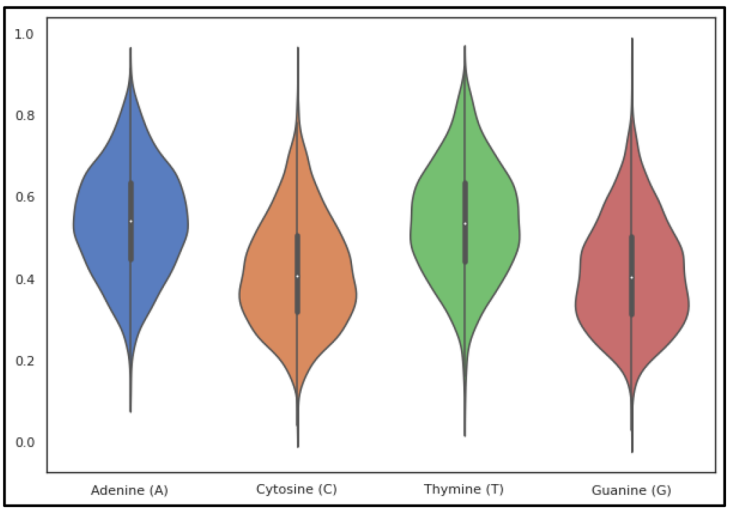

Figure 5. The violin plot of all four nucleotide compositions.

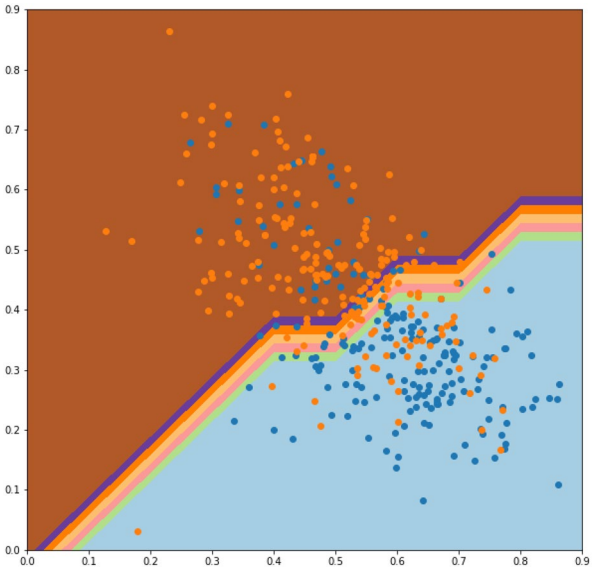

Figure 6. The feature visualization scatter plot of features extracted and used in the proposed study.

features with more robustness to noise and effective against the sensitive DNA Enhancer sites as shown in Fig. 6. All the combined features efficiently differentiate from Enhancers and Non Enhancer sites.

Classification algorithm. *Random forests.* In the past, ensemble learning methods have been applied in many bioinformatics relevant research studies[76,77] and have produced highly efficient outcomes in measures of performance. Ensemble learning methods utilize many classifiers in a classification problem with aggregation of their results. The two most commonly used methods are boosting[78,79] and bagging[80] which perform classifications using trees. In boosting, the trees which are successive, propagate extra weights to points which are predicted incorrectly by the previous classifiers. The weighted vote decides the prediction in the end. Whereas, in bagging, the successive trees do not rely on previous trees, rather, each tree is constructed independently from the data using a bootstrap sample. The simple majority vote decides the prediction in the end.

In bioinformatics and related fields, random forests have grown in popularity as a classification tool. They have also performed admirably in extremely complex data environments. A random sample of the observations, typically a bootstrap sample or a subsample of the original data, is used to build each tree in a random forest. Out-of-bag (OOB) observations are those that are not included in the subsample or the bootstrap sample, respectively. The so-called OOB error can be produced, for instance, by using the OOB observations to estimate the random forest prediction error. The OOB error is frequently used to gauge how well the random forest classifier predicts outcomes and aids in identifying model uncertainties. The OOB error has the benefit of using the entire original sample for both building the random forest classifier and estimating error. In order to add more randomness to bagging, Leo Breiman[81] constructed random forests. The random forests changed the construction of the classification trees by adding the construction of each tree from the data using a different bootstrap sample. The splitting of each node, in standard classification trees, is performed by dividing each node equally among all the variables. However, in random forests, the splitting of each node is performed by choosing the best among a subset of predictors which are chosen randomly at that node (Fig. 7 shows the structure of the random forest classifier). As compared to many other classifiers, such as support vector machine, discriminant analysis and neural networks, this counterintuitive strategy perform very well and is robust against overfitting[76].

Algorithm: supervised learning using random forest. Scikit-Learn[82] library using python was implemented for random forest classifier for fitting the trainings and simulations in our proposed method. The number of trees was increased from the default parameter value of 10 to 100. The number of trees parameter value was optimized to 100 using hyper parameter tuning methods and optimal value for the parameter was searched using the successive halving technique in scikit-learn[82] library. The searching space for the parameter "n_estimators" in random forest classifier was (5–500) which was optimized to 100 after successful halving. One of the key findings observed during the experimentation process was that forest with more than 100 trees minimally contribute to the accuracy of the classifier, but can enhance the overall size of the proposed model substantially. Figure 8a illustrates a flowchart to show the overall process of the proposed method.

Out-of-bag estimation. It is frequently asserted that the OOB error is a neutral estimator of the true error rate. Every observation is "out-of-bag" for some of the trees in a random forest because each tree is constructed from a different sample of the original data. Then, only those trees can be used for which the observation was not used in the construction to derive the prediction for the observation. Each observation is given a classification as a result, and the error rate can be calculated using these predictions. The resulting error rate is referred to as OOB

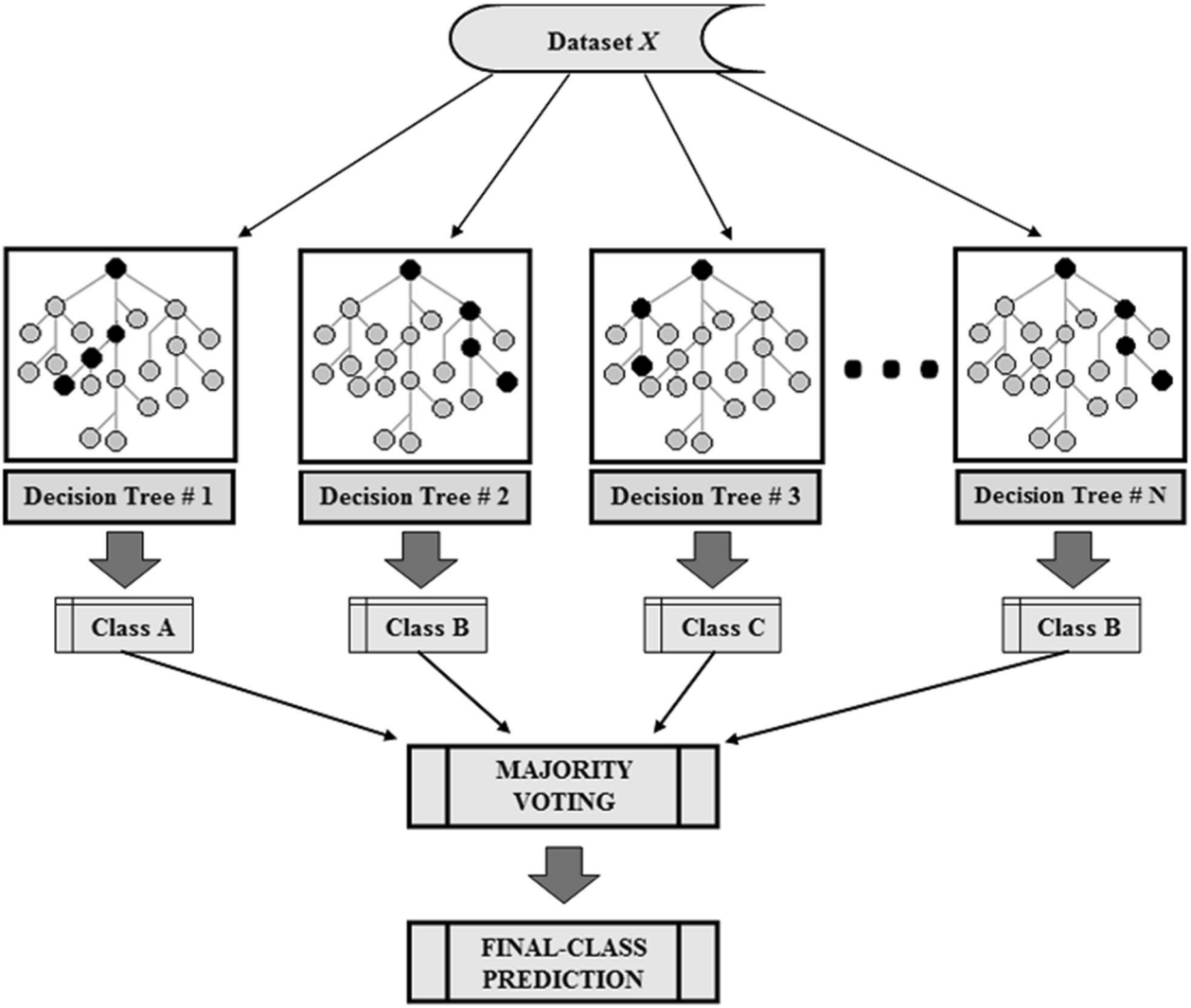

Figure 7. The structure of the random forest classifier.

error. Breiman[81] was the first to propose this process, and it has since gained widespread acceptance as a reliable technique for error estimation in "Random forests". Each new tree is fitted from a bootstrap sample of the training observations $z_i = x_i, y_i$ when training the random forest classifier using bootstrap aggregation. The average error for each z_i calculated using predictions from the trees that do not contain z_i in their respective bootstrap sample is known as the out-of-bag (OOB) error. This makes it possible to fit and validate the random forest classifier while it is being trained. The OOB error is calculated at the addition of each new tree during training, as shown in the plot below. A practitioner can roughly determine the value of n estimators at which the error stabilizes using the resulting Fig. 8b. The scikit-learn[82] library was used to process the out of bag error estimation.

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

Experiments and results

For the assessment and verifications of the model and to analyze its performance, some methods are used to evaluate them. These methods evaluate the classifiers using inspection attributes which are based on the outcomes of classification assessments and estimates.

Cross-validation. *k-fold cross validation.* K-fold cross validation (KFCV) technique is most commonly used by practitioners for estimation of errors in classifications. Also known as rotation estimation, KFCV splits a dataset into 'K' folds which are randomly selected and are equal in size approximately. The prediction error of the fitted model is calculated by predicting the *k*th part of the data which is dependent on other K − 1 parts to fit the model. The error estimates of K from the prediction are combined together using the same procedure for each k = 1, 2, … , K.

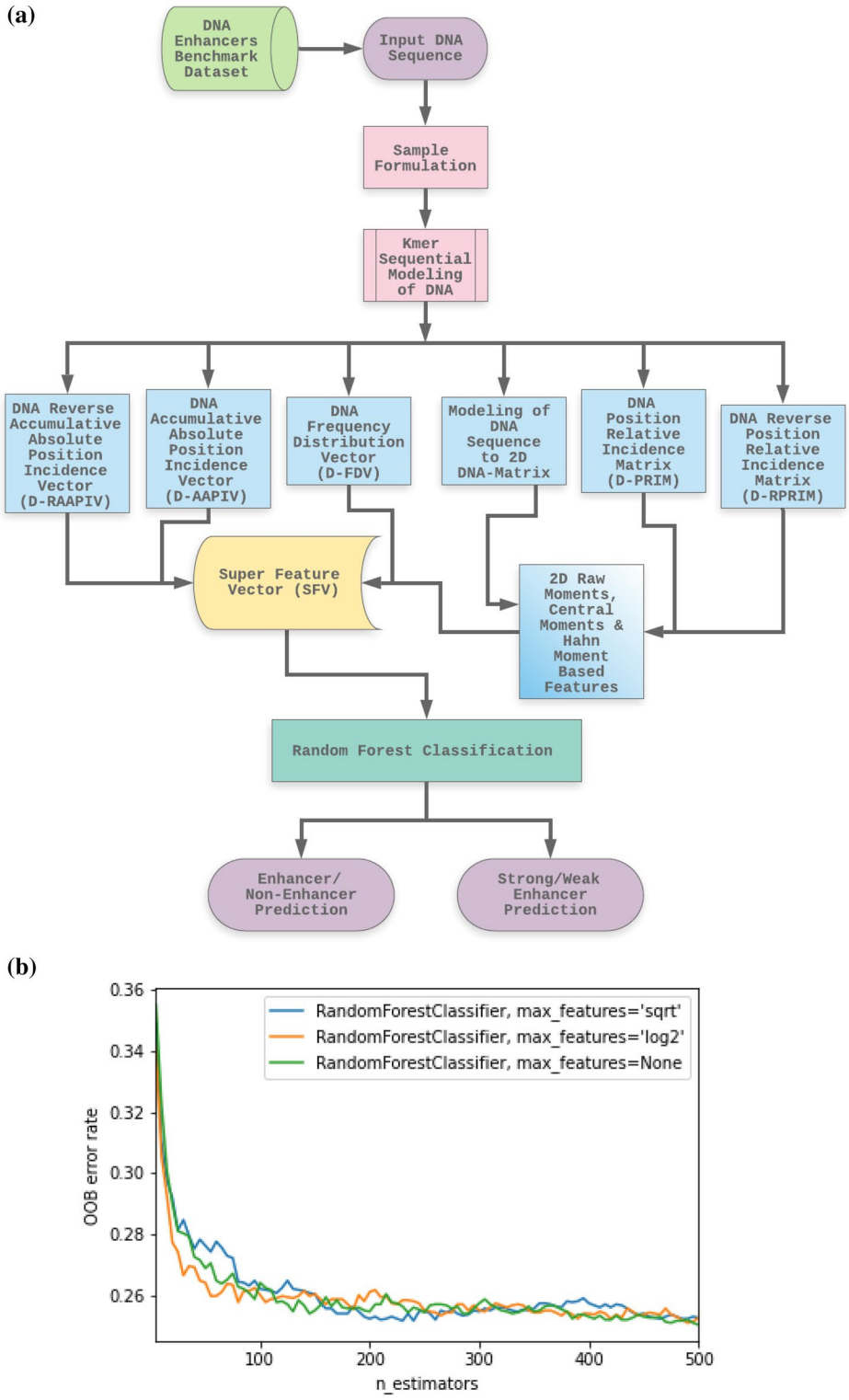

Figure 8. (**a**) The Flowchart of the overall proposed method. (**b**) The OOB error rate stabilization during training estimator trees.

Although the generalization performance of any classifier is mostly estimated using unbiased approximations in jackknife tests, two drawbacks exists in this test, firstly, the variance is high as estimates used in all the datasets are very similar to each other, secondly, its calculative expensive as n estimates are required to be computed,

Symbols	Description of symbols
Y^+	The total number of true enhancers
Y_-^+	The total number of true enhancers incorrectly predicted as non-enhancers
Y^-	The total number of true non-enhancers
Y_+^-	The total number of non-enhancers predicted as enhancers

Table 2. Description of symbols used to define these equations.

and the total number of observations to test is n in the dataset. The fivefold and tenfold cross validation tests are proven to be a good compromise between computational requirements and impartiality.

In the KFCV tests, the selection of 'K' is considered as a significant attribute. To testify errors in prediction models, cross validations (K = 5 and K = 10) tests have been used in many research studies. 5-Fold and 10-Fold tests proved to have accurate results in our proposed model and proved to be much better than state-of-the-art methods. These results are listed in Tables 4, 5 and 6.

Evaluation parameters. The problems of binary classifications use metrics such as Accuracy (Acc), Sensitivity (Sn), Specificity (Sp) and Mathew's Correlation Coefficient (MCC)for measuring the proposed prediction model quality and efficiency. These metrics are defined in the following Eq. (24):

$$\begin{cases} Sn = \frac{TP}{TP+FN} \\ Sp = \frac{TN}{TN+FP} \\ Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \\ MCC = \frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \end{cases} \tag{24}$$

Here true-positives (TP), TN (true-negatives), FP (false-positives) and FN (false-negatives) represent the outcomes from the cross validation tests. Unfortunately, the conventional formulations from the above mentioned metrics in Eq. (24) lack in intuitiveness and due to this fact, understanding these measures especially MCC, many scientists have faced difficulties. To ease this difficulty, the above conventional equations were converted by Xu[83] and Feng[84] using Chou's four intuitive equations which used the symbols introduced by Chou[85]. The symbols that define these equations are; Y^+, Y^-, Y_-^+ and Y_+^-.The description of these symbols is defined in Table 2.

From the above correspondence in Table 2, we can define Eq. (25):

$$\begin{cases} TP = Y^+ - Y_-^+ \\ TN = Y^- - Y_+^- \\ FP = Y_+^- \\ FN = Y_-^+ \end{cases} \tag{25}$$

From the above correspondence in Table 2, we can define Eq. (26):

$$\begin{cases} Sn = 1 - \frac{Y_-^+}{Y^+} \\ Sp = 1 - \frac{Y_+^-}{Y^-} \\ Accuracy = 1 - \frac{Y_-^+ + Y_+^-}{Y^+ + Y^-} \\ MCC = \frac{1-\left(\frac{Y_-^+}{Y^+} + \frac{Y_+^-}{Y^-}\right)}{\sqrt{\left(1+\frac{Y_+^- - Y^+}{Y^+}\right)\left(1+\frac{Y_-^+ - Y_+^-}{Y^-}\right)}} \end{cases} \tag{26}$$

The above Eq. (26) has the same meaning as the Eq. (24) but it is more easy to understand and intuitive. Table 3 defines the detail description of these equations.

The set of metrics used in above Table 3 are not applicable to multi-labeled prediction models rather they are only useful for single labeled-systems. A different set of metrics exists for multi-labeled-systems which have been used by various researchers[86–88]. The comparison of existing classifiers with proposed method is mentioned in Tables 4, 5 and 6.

Results and discussions

The classification algorithms with their predictions results using benchmark dataset are shown in Tables 4, 5 and 6. iEnhancer-EL[89] and iEnhancer-2L[26] produced better outcomes using ensemble classifiers and achieved accuracy of 78.03% and 76.89% respectively in which they were successful in predicting strong enhancers with accuracy of 65.03% and 61.93% respectively. Whereas EnhancerPred[27] achieved 80.82% accuracy and used SVMs which produced slightly better results in predicting strong enhancers with 62.06% accuracy. Similarly, iEnhancer-2L-Hybrid[90] and iEnhancer-5Step[29] improved the accuracy results with their prediction model and acquired77.86% and 82.3% accuracies respectively with identifying the strong enhancers with 65.83% and 68.1% accuracies respectively. In contrast, 91.68%and 84.53%accuracy was achieved in predicting enhancers and their

When	Then	Description
$Y_{-}^{+}=0$	Sn = 1	None of the enhancer is predicted as a non-enhancer
$Y_{-}^{+}=Y^{+}$	Sn = 0	All of the enhancers were incorrectly predicted as non-enhancers
$Y_{+}^{-}=0$	Sp = 1	None of the non-enhancer is incorrectly predicted as an enhancer
$Y_{+}^{-}=Y^{-}$	Sp = 0	All of the non-enhancers are incorrectly predicted as enhancers
$Y_{-}^{+}+Y_{+}^{-}=0$	ACC = 1, MCC = 1	None of the enhancers and none of the non-enhancers were incorrectly predicted
$Y_{-}^{+}=Y^{+}$ and $Y_{+}^{-}=Y^{-}$	ACC = 0, MCC = −1	All of the enhancers and all of the non-enhancers were incorrectly predicted
$Y_{-}^{+}=\frac{Y^{+}}{2}$ and $Y_{+}^{-}=\frac{Y^{-}}{2}$	ACC = 0.5, MCC = 0	The overall prediction is not a better than any other random prediction outcome

Table 3. Description of equations used Eqs. (26).

Layer	Classifiers	Sn (%)	Sp (%)	ACC (%)	MCC	AUC
1	iEnhancer-2L	78.09	75.88	76.89	0.54	0.85
	iEnhancer-2L-Hybrid	75.33	80.39	77.86	0.558	–
	EnhancerPred	72.57	73.79	77.39	0.464	–
	iEnhancer-EL	75.67	80.39	78.03	0.561	–
	iEnhancer-5Step	81.1	83.5	82.3	0.65	–
	Proposed method	84.90	88.21	86.56	0.7319	0.93
2	iEnhancer-2L	62.21	61.82	61.93	0.24	0.66
	iEnhancer-2L-Hybrid	71.02	60.64	65.83	0.318	–
	EnhancerPred	62.67	61.46	68.19	0.2413	–
	iEnhancer-EL	69	61.05	65.03	0.315	–
	iEnhancer-5Step	75.3	60.8	68.1	0.37	–
	ES-ARCNN	72.78	59.57	66.17	0.3263	–
	Proposed method	81.54	63.06	72.30	0.4537	0.80

Table 4. Comparison of state-of-the-art methods with the proposed method using 5-fold cross validation tests.

Layer	Classifier	Sn(%)	Sp(%)	ACC(%)	MCC	AUC	AUPR
1	KNN	69.81	72.90	71.36	0.4275	0.89	0.80
	Naïve Bayes	67.59	69.47	68.53	0.3712	0.78	0.72
	AdaBoost	72.30	73.31	72.80	0.4569	0.89	0.80
	SVM	70.68	78.43	74.56	0.4933	0.84	0.82
	Probalistic NN	72.04	72.97	72.51	0.4507	0.81	0.84
	Random forest	86.53	96.90	91.72	0.8398	0.87	0.97
2	KNN	58.77	54.05	56.41	0.1285	0.58	0.57
	Naïve Bayes	58.35	59.56	58.95	0.1793	0.62	0.61
	AdaBoost	63.46	57.15	60.31	0.2079	0.63	0.66
	SVM	69.94	55.68	62.80	0.2598	0.66	0.66
	Probalistic NN	76.95	44.34	60.64	0.2261	0.64	0.69
	Random forest	80.49	93.97	87.23	0.7519	0.82	0.93

Table 5. Comparison of classifiers for predicting enhancers using tenfold cross validations.

strength respectively by the currently proposed method after utilizing obscure features from statistical moments and random forest classifications using 5-Fold cross validation tests (see Table 4 and Fig. 9 for ROCs). Furthermore, tenfold cross-validation test was also conducted using random forest classifier on benchmark dataset and obtained the accuracy results are listed in Table 5. The ROCs of 10-fold cross-validation tests are shown in Figs. 10 and 11. The violin plots of 5 fold cross-validation tests are shown in Fig. 12. In addition to cross validation tests, an independent test was also performed using the independent dataset. The comparison of proposed model and state-of-the-art methods using independent dataset is listed in Table 6 and ROC is shown in Fig. 13. Furthermore, jackknife test was also performed on these datasets. A detailed comparison of some selected machine learning algorithms using jackknife test is mentioned in Table 7. The Precision-Recall (PR) curves for enhancer and their strength prediction is also labeled in Figs. 14 and 15 respectively. The proposed method is based on the feature sets that are evaluated using Hahn moments which are easier for the random forest based classifier

Layer	Classifiers	Sn (%)	Sp (%)	ACC (%)	MCC	AUC
1	iEnhancer-2L	71	75	73	0.46	0.80
	EnhancerPred	73.5	74.5	74	0.48	0.81
	iEnhancer-EL	71	78.5	74.75	0.496	0.82
	iEnhancer-5Step	82	76	79	0.58	0.87
	iEnhancer-ECNN	78.5	75.2	76.9	0.537	0.83
	iEnhancer-RD	81.0	76.5	78.8	0.576	0.84
	Proposed method	78.10	81.05	79.50	0.5907	0.93
2	iEnhancer-2L	47	74	60.5	0.2181	–
	EnhancerPred	45	65	55	0.1021	–
	iEnhancer-EL	54	68	61	0.2222	–
	iEnhancer-5Step	74	53	63.5	0.28	–
	iEnhancer-ECNN	79.1	56.4	67.8	0.368	–
	iEnhancer-RD	84.0	57.0	70.5	0.426	–
	ES-ARCNN	86	45	65.5	0.3399	–
	Proposed method	68.29	79.22	72.5	0.4624	–

Table 6. Independent tests based comparison of state-of-the-art methods with the proposed method.

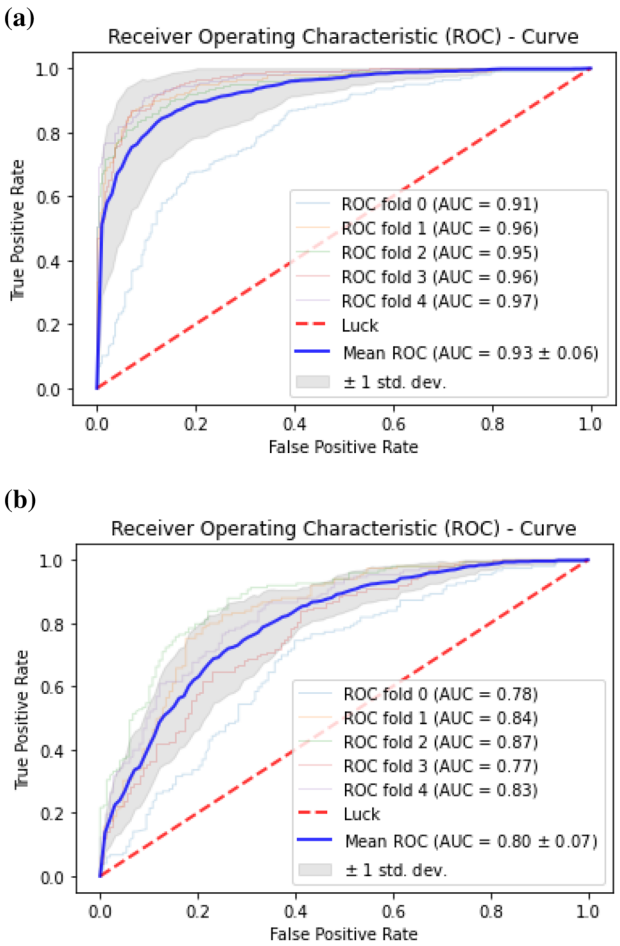

Figure 9. (**a**) ROC curve of fivefold cross validation tests for enhancers. (**b**) ROC curve of fivefold cross validation tests for enhancer strengths.

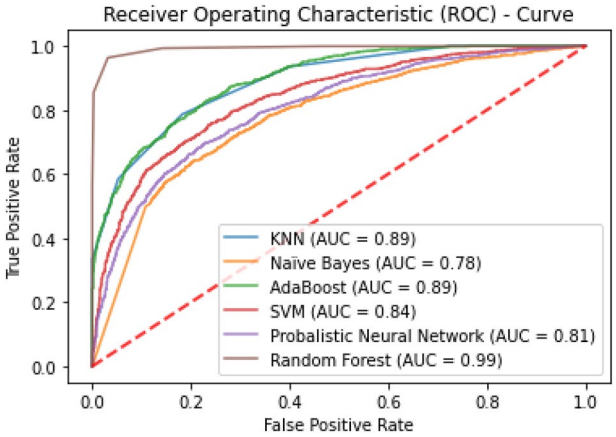

Figure 10. 10 fold test ROCs comparison of classifiers for enhancer site prediction.

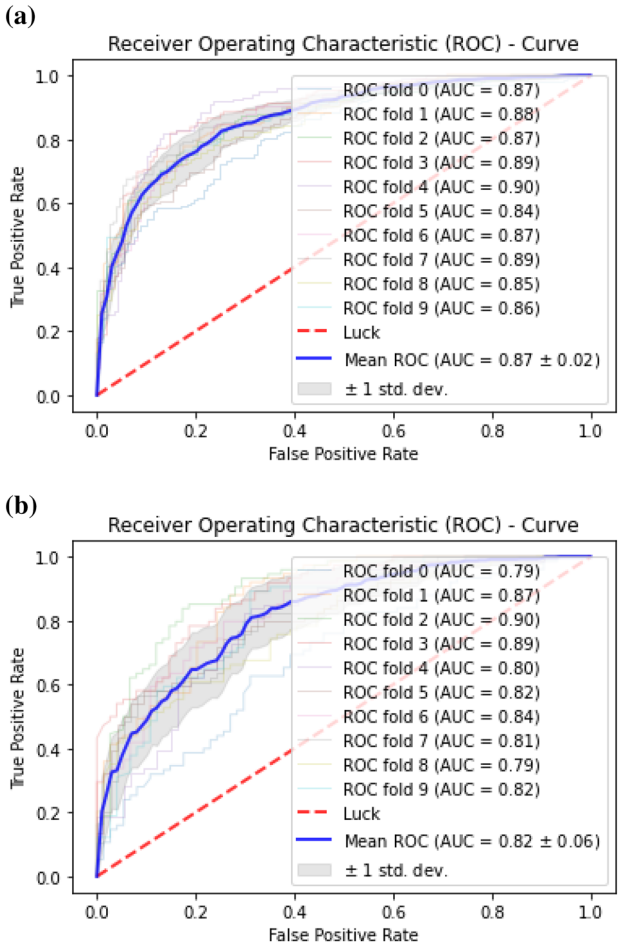

Figure 11. (**a**) ROC curve of tenfold cross validation tests for enhancers using (random forest). (**b**) ROC curve of tenfold cross validation tests for enhancers strength (random forest).

to classify the feature vectors in acute time and are very efficient as compared to previous methods which were not able to produce better results on the computational cost of training and testing using classification process.

Web-server. As observed in past studies[91–95], the development of a web-server is highly significant and useful for building more useful prediction methodologies. Thus, efforts for a user friendly webserver have been made in past[72,96–99] to provide ease to biologists and scientists in drug discovery. The software code which has

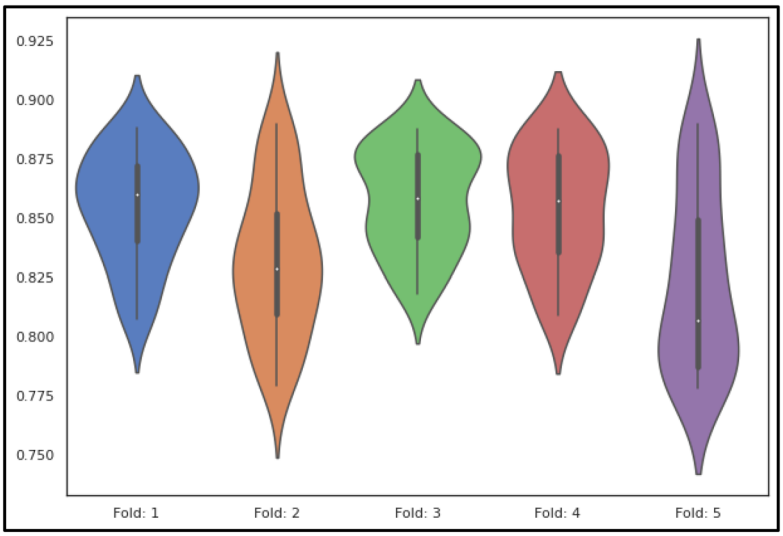

Figure 12. Violinplot of fivefold cross validation for enhancers (random forest).

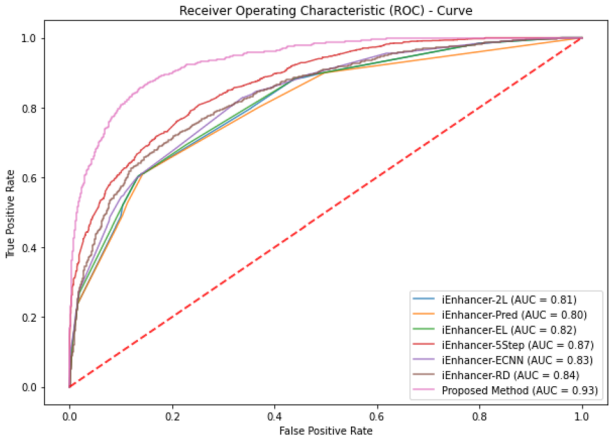

Figure 13. The ROCs of state of art methods using independent tests for enhancer prediction.

Layer	Classifier	Sn (%)	Sp (%)	ACC (%)	AUC
1	KNN	70.89	78.77	74.83	0.86
	Naïve Bayes	67.58	69.33	68.46	0.79
	Gaussian Naïve Bayes	71.63	71.09	71.36	0.90
	Random forest	75.26	97.43	86.35	0.95
2	KNN	70.35	53.23	61.79	0.76
	Naïve Bayes	57.95	59.43	58.69	0.67
	Gaussian Naïve Bayes	69.67	52.02	60.84	0.69
	Random forest	68.86	97.17	83.01	0.92

Table 7. Jackknife test comparison of machine learning algorithms for predicting enhancers and their strengths.

been developed for the proposed method is accessible at https://github.com/csbioinfopk/enpred which is developed using Python, Scikit-Learn and Flask. The webserver to the current study will be provided for the research community in near future.

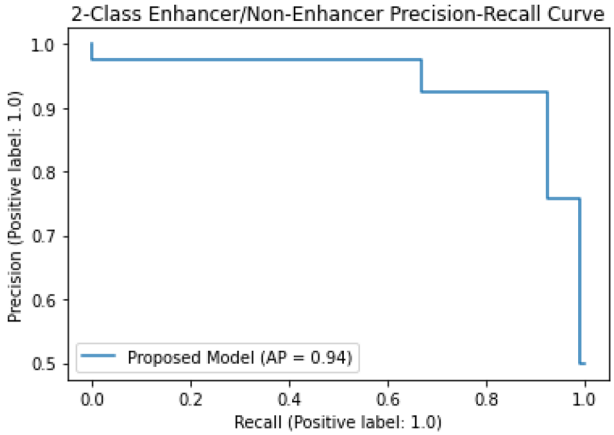

Figure 14. The PR curves of random forest using jackknife tests for enhancer site prediction.

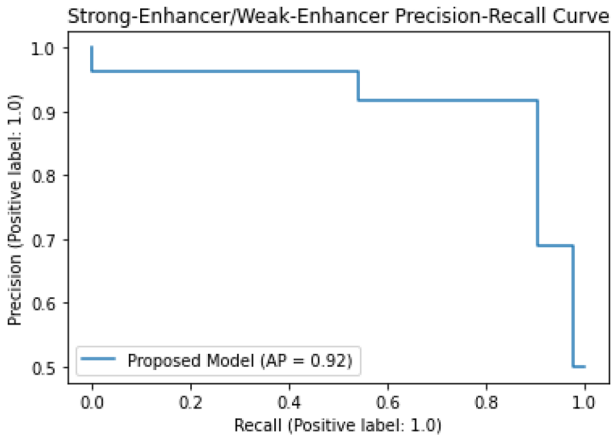

Figure 15. The PR curves of random forest using jackknife tests for enhancer strength prediction.

Conclusion

In the proposed research, an efficient model for predicting the enhancers and their strength using statistical moments and random forest classifier is developed. In recent past, many methods were proposed to predict enhancers, but our method has proved to be better in accuracy than the existing state-of-the-art methods. Our method achieved accuracies of91.68% and 84.53% for enhancer and strong enhancer classifications using 5 Fold tests on a benchmark dataset which is currently the highest and accurate classification method for prediction of enhancers and their strength.

Data availability

The Online Supporting Information S1 (https://github.com/csbioinfopk/enpred/blob/master/static/Supp-S1.pdf) provides sequence information of DNA Enhancer and non-Enhancer sites used for training, Online Supporting Information S2 (https://github.com/csbioinfopk/enpred/blob/master/static/Supp-S2.pdf) provides sequence information of DNA Strong enhancer sites and Weak enhancer sites, and Online Supporting Information S3 (https://github.com/csbioinfopk/enpred/blob/master/static/Supp-S3.pdf) provides sequence information of DNA Sample sequences used for Independent Tests. The GitHub repository provide access to all the data necessary with relevant accession numbers to substantiate the study's findings.

Received: 28 March 2022; Accepted: 24 August 2022
Published online: 07 September 2022

References

1. Erwin, G. D. *et al.* Integrating diverse datasets improves developmental enhancer prediction. *PLoS Comput. Biol.* **10**(6), e1003677–e1003677. https://doi.org/10.1371/journal.pcbi.1003677 (2014).
2. Visel, A., Rubin, E. M. & Pennacchio, L. A. Genomic views of distant-acting enhancers. *Nature* **461**(7261), 199–205. https://doi.org/10.1038/nature08451 (2009).
3. Sakabe, N. J., Savic, D. & Nobrega, M. A. Transcriptional enhancers in development and disease. *Genome Biol.* **13**(1), 238 (2012).

4. Heintzman, N. D. & Ren, B. Finding distal regulatory elements in the human genome. *Curr. Opin. Genet. Dev.* **19**(6), 541–549. https://doi.org/10.1016/j.gde.2009.09.006 (2009).
5. Blackwood, E. M. & Kadonaga, J. T. Going the distance: A current view of enhancer action. *Science* **281**, 60 (1998).
6. Pennacchio, L. A., Bickmore, W., Dean, A., Nobrega, M. A. & Bejerano, G. Enhancers: Five essential questions. *Nat. Rev. Genet.* **14**, 288 (2013).
7. Kulaeva, O. I., Nizovtseva, E. V., Polikanov, Y. S., Ulianov, S. V. & Studitsky, V. M. Distant activation of transcription: Mechanisms of enhancer action. *Mol. Cell. Biol.* **32**(24), 4892–4897. https://doi.org/10.1128/mcb.01127-12 (2012).
8. Herz, H.-M. Enhancer deregulation in cancer and other diseases. *BioEssays* **38**(10), 1003–1015. https://doi.org/10.1002/bies.201600106 (2016).
9. Zhang, G. *et al.* DiseaseEnhancer: A resource of human disease-associated enhancer catalog. *Nucleic Acids Res.* **46**(D1), D78–D84. https://doi.org/10.1093/nar/gkx920 (2017).
10. Corradin, O. & Scacheri, P. C. Enhancer variants: Evaluating functions in common disease. *Genome Med.* **6**, 85 (2014).
11. Boyd, M. *et al.* Characterization of the enhancer and promoter landscape of inflammatory bowel disease from human colon biopsies. *Nat. Commun.* **9**, 1661 (2018).
12. Banerji, J., Rusconi, S. & Schaffner, W. Expression of a β-globin gene is enhanced by remote SV40 DNA sequences. *Cell* **27**(2), 299–308 (1981).
13. Shlyueva, D., Stampfel, G. & Stark, A. Transcriptional enhancers: From properties to genome-wide predictions. *Nat. Rev. Genet.* **15**(4), 272–286. https://doi.org/10.1038/nrg3682 (2014).
14. Heintzman, N. D. *et al.* Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. *Nat. Genet.* **39**(3), 311 (2007).
15. Jin, F., Li, Y., Ren, B. & Natarajan, R. PU. 1 and C/EBPα synergistically program distinct response to NF-κB activation through establishing monocyte specific enhancers. *Proc. Natl. Acad. Sci.* **108**(13), 5290–5295 (2011).
16. Kim, T.-K. *et al.* Widespread transcription at neuronal activity-regulated enhancers. *Nature* **465**(7295), 182 (2010).
17. Boyle, A. P. *et al.* High-resolution genome-wide in vivo footprinting of diverse transcription factors in human cells. *Genome Res.* **21**(3), 456–464. https://doi.org/10.1101/gr.112656.110 (2011).
18. Visel, A. *et al.* ChIP-seq accurately predicts tissue-specific activity of enhancers. *Nature* **457**(7231), 854–858. https://doi.org/10.1038/nature07730 (2009).
19. Ernst, J. *et al.* Mapping and analysis of chromatin state dynamics in nine human cell types. *Nature* **473**(7345), 43–49. https://doi.org/10.1038/nature09906 (2011).
20. Fernández, M. & Miranda-Saavedra, D. Genome-wide enhancer prediction from epigenetic signatures using genetic algorithm-optimized support vector machines. *Nucleic Acids Res.* **40**(10), e77–e77. https://doi.org/10.1093/nar/gks149 (2012).
21. Firpi, H. A., Ucar, D. & Tan, K. Discover regulatory DNA elements using chromatin signatures and artificial neural network. *Bioinformatics* **26**(13), 1579–1586. https://doi.org/10.1093/bioinformatics/btq248 (2010).
22. Kleftogiannis, D., Kalnis, P. & Bajic, V. B. DEEP: A general computational framework for predicting enhancers. *Nucleic Acids Res.* **43**(1), e6. https://doi.org/10.1093/nar/gku1058 (2015).
23. Rajagopal, N. *et al.* RFECS: A random-forest based algorithm for enhancer identification from chromatin state. *PLoS Comput. Biol.* **9**(3), e1002968–e1002968. https://doi.org/10.1371/journal.pcbi.1002968 (2013).
24. Bu, H., Gan, Y., Wang, Y., Zhou, S. & Guan, J. A new method for enhancer prediction based on deep belief network. *BMC Bioinform.* **18**, 418 (2017).
25. Yang, B. *et al.* BiRen: Predicting enhancers with a deep-learning-based model using the DNA sequence alone. *Bioinformatics* **33**(13), 1930–1936 (2017).
26. Liu, B., Fang, L., Long, R., Lan, X. & Chou, K. C. iEnhancer-2L: A two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition. *Bioinformatics* **32**(3), 362–369. https://doi.org/10.1093/bioinformatics/btv604 (2016).
27. Jia, C. & He, W. EnhancerPred: A predictor for discovering enhancers based on the combination and selection of multiple features. *Sci. Rep.* https://doi.org/10.1038/srep38741 (2016).
28. He, W. & Jia, C. EnhancerPred2.0: Predicting enhancers and their strength based on position-specific trinucleotide propensity and electron-ion interaction potential feature selection. *Mol. BioSyst.* **13**(4), 767–774. https://doi.org/10.1039/c7mb00054e (2017).
29. Le, N. Q. K. *et al.* iEnhancer-5Step: Identifying enhancers using hidden information of DNA sequences via Chou's 5-step rule and word embedding. *Anal. Biochem.* **571**, 53–61. https://doi.org/10.1016/j.ab.2019.02.017 (2019).
30. Yang, H., Wang, S. & Xia, X. iEnhancer-RD: Identification of enhancers and their strength using RKPK features and deep neural networks. *Anal. Biochem.* **630**, 114318. https://doi.org/10.1016/j.ab.2021.114318 (2021).
31. Zhang, T.-H., Flores, M. & Huang, Y. ES-ARCNN: Predicting enhancer strength by using data augmentation and residual convolutional neural network. *Anal. Biochem.* **618**, 114120. https://doi.org/10.1016/j.ab.2021.114120 (2021).
32. Yang, R., Wu, F., Zhang, C. & Zhang, L. iEnhancer-GAN: A deep learning framework in combination with word embedding and sequence generative adversarial net to identify enhancers and their strength. *Int. J. Mol. Sci.* **22**(7), 3589. https://doi.org/10.3390/ijms22073589 (2021).
33. Cai, L. *et al.* iEnhancer-XG: Interpretable sequence-based enhancers and their strength predictor. *Bioinformatics* **37**(8), 1060–1067. https://doi.org/10.1093/bioinformatics/btaa914 (2021).
34. Lyu, Y. *et al.* iEnhancer-KL: A novel two-layer predictor for identifying enhancers by position specific of nucleotide composition. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **18**(6), 2809–2815. https://doi.org/10.1109/TCBB.2021.3053608 (2021).
35. Le, N. Q. K., Ho, Q.-T., Nguyen, T.-T.-D. & Ou, Y.-Y. 'A transformer architecture based on BERT and 2D convolutional neural network to identify DNA enhancers from sequence information. *Brief Bioinform.* **22**(5), bbab005. https://doi.org/10.1093/bib/bbab005 (2021).
36. Liang, Y., Zhang, S., Qiao, H. & Cheng, Y. iEnhancer-MFGBDT: Identifying enhancers and their strength by fusing multiple features and gradient boosting decision tree. *Math. Biosci. Eng.* **18**(6), 8797–8814. https://doi.org/10.3934/mbe.2021434 (2021).
37. Nguyen, Q. H. *et al.* iEnhancer-ECNN: Identifying enhancers and their strength using ensembles of convolutional neural networks. *BMC Genomics* **20**(Suppl 9), 951. https://doi.org/10.1186/s12864-019-6336-3 (2019).
38. Tan, K. K., Le, N. Q. K., Yeh, H. Y. & Chua, M. C. H. 'Ensemble of deep recurrent neural networks for identifying enhancers via dinucleotide physicochemical properties. *Cells* **8**(7), 767. https://doi.org/10.3390/cells8070767 (2019).
39. Fu, L., Niu, B., Zhu, Z., Wu, S. & Li, W. CD-HIT: Accelerated for clustering the next-generation sequencing data. *Bioinformatics* **28**(23), 3150–3152. https://doi.org/10.1093/bioinformatics/bts565 (2012).
40. Chou, K.-C. Impacts of bioinformatics to medicinal chemistry. *Med. Chem.* **11**(3), 218–234. https://doi.org/10.2174/1573406411666141229162834 (2015).
41. Chou, K. C. Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins Struct. Funct. Genet.* **43**(3), 246–255. https://doi.org/10.1002/prot.1035 (2001).
42. Cao, D.-S., Xu, Q.-S. & Liang, Y.-Z. propy: A tool to generate various modes of Chou's PseAAC. *Bioinformatics* **29**(7), 960–962. https://doi.org/10.1093/bioinformatics/btt072 (2013).
43. Du, P., Wang, X., Xu, C. & Gao, Y. PseAAC-Builder: A cross-platform stand-alone program for generating various special Chou's pseudo-amino acid compositions. *Anal. Biochem.* **425**, 117–119. https://doi.org/10.1016/j.ab.2012.03.015 (2012).
44. Du, P., Gu, S. & Jiao, Y. PseAAC-general: Fast building various modes of general form of Chou's pseudo-amino acid composition for large-scale protein datasets. *Int. J. Mol. Sci.* **15**, 3495 (2014).

45. Chou, K. C. Some remarks on protein attribute prediction and pseudo amino acid composition. *J. Theor. Biol.* **273**(1), 236–247. https://doi.org/10.1016/j.jtbi.2010.12.024 (2011).
46. Chou, K.-C. Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology. *Curr. Proteom.* **6**(4), 262–274. https://doi.org/10.2174/157016409789973707 (2009).
47. Chen, W., Lei, T. Y., Jin, D. C., Lin, H. & Chou, K. C. PseKNC: A flexible web server for generating pseudo K-tuple nucleotide composition. *Anal. Biochem.* **456**(1), 53–60. https://doi.org/10.1016/j.ab.2014.04.001 (2014).
48. Chen, W., Lin, H. & Chou, K. C. Pseudo nucleotide composition or PseKNC: An effective formulation for analyzing genomic sequences. *Mol. BioSyst.* **11**(10), 2620–2634. https://doi.org/10.1039/c5mb00155b (2015).
49. Liu, B., Yang, F., Huang, D.-S. & Chou, K.-C. iPromoter-2L: A two-layer predictor for identifying promoters and their types by multi-window-based PseKNC. *Bioinformatics* **34**(1), 33–40. https://doi.org/10.1093/bioinformatics/btx579 (2017).
50. Liu, B. *et al.* 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. https://doi.org/10.1093/nar/gkv458 (2015).
51. Liu, B., Wu, H. & Chou, K.-C. 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.* **09**(04), 67–91. https://doi.org/10.4236/ns.2017.94007 (2017).
52. Liu, B., Long, R. & Chou, K. C. IDHS-EL: Identifying DNase i hypersensitive sites by fusing three different modes of pseudo nucleotide composition into an ensemble learning framework. *Bioinformatics* **32**(16), 2411–2418. https://doi.org/10.1093/bioinformatics/btw186 (2016).
53. Papademetriou, R. C. 'Reconstructing with moments. *Proc. Int. Conf. Pattern Recogn.* **3**, 476–480. https://doi.org/10.1109/ICPR.1992.202028 (1992).
54. Butt, A. H., Khan, S. A., Jamil, H., Rasool, N. & Khan, Y. D. A prediction model for membrane proteins using moments based features. *Biomed. Res. Int.* **2016**, 1–7. https://doi.org/10.1155/2016/8370132 (2016).
55. Butt, A. H., Rasool, N. & Khan, Y. D. A treatise to computational approaches towards prediction of membrane protein and its subtypes. *J. Membr. Biol.* **250**(1), 55–76. https://doi.org/10.1007/s00232-016-9937-7 (2017).
56. Butt, A. H., Rasool, N. & Khan, Y. D. Predicting membrane proteins and their types by extracting various sequence features into Chou's general PseAAC. *Mol. Biol. Rep.* **45**(6), 2295–2306. https://doi.org/10.1007/s11033-018-4391-5 (2018).
57. Butt, A. H., Rasool, N. & Khan, Y. D. Prediction of antioxidant proteins by incorporating statistical moments based features into Chou's PseAAC. *J. Theor. Biol.* **473**, 1–8. https://doi.org/10.1016/j.jtbi.2019.04.019 (2019).
58. Butt, A. H. & Khan, Y. D. CanLect-Pred: A cancer therapeutics tool for prediction of target cancer lectins using experiential annotated proteomic sequences. *IEEE Access* https://doi.org/10.1109/ACCESS.2019.2962002 (2020).
59. Khan, Y. D., Khan, N. S., Naseer, S. & Butt, A. H. iSUMOK-PseAAC: Prediction of lysine sumoylation sites using statistical moments and Chou's PseAAC. *PeerJ* **9**, e11581. https://doi.org/10.7717/peerj.11581 (2021).
60. Khan, S. A., Khan, Y. D., Ahmad, S. & Allehaibi, K. H. *N*-MyristoylG-PseAAC: Sequence-based prediction of *N*-Myristoyl glycine sites in proteins by integration of PseAAC and statistical moments. *Lett. Org. Chem.* **16**(3), 226–234. https://doi.org/10.2174/1570178616666181217153958 (2019).
61. Amanat, S., Ashraf, A., Hussain, W., Rasool, N. & Khan, Y. D. Identification of lysine carboxylation sites in proteins by integrating statistical moments and position relative features via general PseAAC. *Curr. Bioinform.* **15**(5), 396–407. https://doi.org/10.2174/1574893614666190723114923 (2020).
62. Mahmood, M. K., Ehsan, A., Khan, Y. D. & Chou, K.-C. iHyd-LysSite (EPSV): Identifying hydroxylysine sites in protein using statistical formulation by extracting enhanced position and sequence variant feature technique. *Curr. Genomics* **21**(7), 536–545. https://doi.org/10.2174/1389202921999200831142629 (2020).
63. Khan, Y. D., Khan, S. A., Ahmad, F. & Islam, S. Iris recognition using image moments and k-Means algorithm. *Sci. World J.* **2014**, 1–9. https://doi.org/10.1155/2014/723595 (2014).
64. Zhou, J., Shu, H., Zhu, H., Toumoulin, C., & Luo, L. Image analysis by discrete orthogonal Hahn moments. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 3656. 524–531. https://doi.org/10.1007/11559573_65 (LNCS, 2005).
65. Zhu, H., Shu, H., Zhou, J., Luo, L. & Coatrieux, J. L. Image analysis by discrete orthogonal dual Hahn moments. *Pattern Recogn. Lett.* **28**(13), 1688–1704. https://doi.org/10.1016/j.patrec.2007.04.013 (2007).
66. Yap, P. T., Paramesran, R. & Ong, S. H. Image analysis using Hahn moments. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(11), 2057–2062. https://doi.org/10.1109/TPAMI.2007.70709 (2007).
67. Goh, H.-A., Chong, C.-W., Besar, R., Abas, F. S. & Sim, K.-S. Translation and scale invariants of Hahn moments. *Int. J. Image Graph.* **09**(02), 271–285. https://doi.org/10.1142/s0219467809003435 (2009).
68. Alghamdi, W., Alzahrani, E., Ullah, M. Z. & Khan, Y. D. 4mC-RF: Improving the prediction of 4mC sites using composition and position relative features and statistical moment. *Anal. Biochem.* **633**, 114385. https://doi.org/10.1016/j.ab.2021.114385 (2021).
69. Malebary, S. J., ur Rehman, M. S. & Khan, Y. D. iCrotoK-PseAAC: Identify lysine crotonylation sites by blending position relative statistical features according to the Chou's 5-step rule. *PLoS ONE* **14**(11), 0223993. https://doi.org/10.1371/journal.pone.0223993 (2019).
70. Shah, A. A. & Khan, Y. D. Identification of 4-carboxyglutamate residue sites based on position based statistical feature and multiple classification. *Sci. Rep.* **10**(1), 16913. https://doi.org/10.1038/s41598-020-73107-y (2020).
71. Ilyas, S. *et al.* iMethylK_pseAAC: Improving accuracy of lysine methylation sites identification by incorporating statistical moments and position relative features into general PseAAC via Chou's 5-steps rule. *Curr. Genom.* **20**(4), 275–292. https://doi.org/10.2174/1389202920666190809095206 (2019).
72. Awais, M. *et al.* iPhosH-PseAAC: Identify phosphohistidine sites in proteins by blending statistical moments and position relative features according to the Chou's 5-step rule and general pseudo amino acid composition. *IEEE/ACM Trans. Comput. Biol. Bioinform.* https://doi.org/10.1109/TCBB.2019.2919025 (2019).
73. Barukab, O., Khan, Y. D., Khan, S. A. & Chou, K.-C. iSulfoTyr-PseAAC: Identify tyrosine sulfation sites by incorporating statistical moments via Chou's 5-steps rule and pseudo components. *Curr. Genomics* **20**(4), 306–320. https://doi.org/10.2174/1389202920666190819091609 (2019).
74. Akmal, M. A., Rasool, N. & Khan, Y. D. Prediction of N-linked glycosylation sites using position relative features and statistical moments. *PLoS ONE* **12**(8), e0181966–e0181966. https://doi.org/10.1371/journal.pone.0181966 (2017).
75. Khan, Y. D., Batool, A., Rasool, N., Khan, S. A. & Chou, K.-C. Prediction of nitrosocysteine sites using position and composition variant features. *Lett. Org. Chem.* **16**(4), 283–293. https://doi.org/10.2174/1570178615666180802122953 (2018).
76. Tyryshkina, A., Coraor, N. & Nekrutenko, A. Predicting runtimes of bioinformatics tools based on historical data: Five years of Galaxy usage. *Bioinformatics* **35**(18), 3453–3460. https://doi.org/10.1093/bioinformatics/btz054 (2019).
77. Simidjievski, N., Todorovski, L. & Džeroski, S. Modeling dynamic systems with efficient ensembles of process-based models. *PLoS ONE* **11**, 4. https://doi.org/10.1371/journal.pone.0153507 (2016).
78. Freund, Y. & Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Lect. Notes Comput. Sci. (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **904**(1), 23–37. https://doi.org/10.1006/jcss.1997.1504 (1995).
79. Schapire, R. E. Theoretical, views of boosting and applications. *Lect. Notes Comput. Sci. (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **1720**, 13–25. https://doi.org/10.1007/3-540-46769-6_2 (1999).
80. Breiman, L. Bagging predictors. *Mach. Learn.* **24**(2), 123–140. https://doi.org/10.1007/bf00058655 (1996).

81. Breiman, L. Random forests. *Mach. Learn.* **45**(1), 5–32. https://doi.org/10.1023/A:1010933404324 (2001).
82. Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
83. Xu, Y., Shao, X. J., Wu, L. Y., Deng, N. Y. & Chou, K. C. ISNO-AAPair: Incorporating amino acid pairwise coupling into PseAAC for predicting cysteine S-nitrosylation sites in proteins. *PeerJ* **2013**(1), e171–e171. https://doi.org/10.7717/peerj.171 (2013).
84. Feng, P. M., Ding, H., Chen, W. & Lin, H. Naïve bayes classifier with feature selection to identify phage virion proteins. *Comput. Math. Methods Med.* **2013**, 1–6. https://doi.org/10.1155/2013/530696 (2013).
85. Chou, K. C. Prediction of signal peptides using scaled window. *Peptides* **22**(12), 1973–1979. https://doi.org/10.1016/S0196-9781(01)00540-X (2001).
86. Xiao, X., Wang, P., Lin, W. Z., Jia, J. H. & Chou, K. C. IAMP-2L: A two-level multi-label classifier for identifying antimicrobial peptides and their functional types. *Anal. Biochem.* **436**(2), 168–177. https://doi.org/10.1016/j.ab.2013.01.019 (2013).
87. Xiao, X., Wu, Z. C. & Chou, K. C. iLoc-Virus: A multi-label learning classifier for identifying the subcellular localization of virus proteins with both single and multiple sites. *J. Theor. Biol.* **284**(1), 42–51. https://doi.org/10.1016/j.jtbi.2011.06.005 (2011).
88. Lin, W. Z., Fang, J. A., Xiao, X. & Chou, K. C. ILoc-Animal: A multi-label learning classifier for predicting subcellular localization of animal proteins. *Mol. BioSyst.* **9**(4), 634–644. https://doi.org/10.1039/c3mb25466f (2013).
89. Liu, B., Li, K., Huang, D. S. & Chou, K. C. IEnhancer-EL: Identifying enhancers and their strength with ensemble learning approach. *Bioinformatics* **34**(22), 3835–3842. https://doi.org/10.1093/bioinformatics/bty458 (2018).
90. Tahir, M., Hayat, M. & Khan, S. A. A two-layer computational model for discrimination of enhancer and their types using hybrid features pace of pseudo K-tuple nucleotide composition. *Arab. J. Sci. Eng.* **43**(12), 6719–6727. https://doi.org/10.1007/s13369-017-2818-2 (2018).
91. Cheng, X., Xiao, X. & Chou, K. C. pLoc_bal-mGneg: Predict subcellular localization of Gram-negative bacterial proteins by quasi-balancing training dataset and general PseAAC. *J. Theor. Biol.* **458**, 92–102. https://doi.org/10.1016/j.jtbi.2018.09.005 (2018).
92. Chou, K.-C. Proposing pseudo amino acid components is an important milestone for proteome and genome analyses. *Int. J. Pept. Res. Ther.* https://doi.org/10.1007/s10989-019-09910-7 (2019).
93. Liu, B., Wu, H., Zhang, D., Wang, X. & Chou, K. C. Pse-Analysis: A python package for DNA/RNA and protein/ peptide sequence analysis based on pseudo components and kernel methods. *Oncotarget* **8**(8), 13338–13343. https://doi.org/10.18632/oncotarget.14524 (2017).
94. Liu, Z. *et al.* pRNAm-PC: Predicting N6-methyladenosine sites in RNA sequences via physical-chemical properties. *Anal. Biochem.* **497**, 60–67. https://doi.org/10.1016/j.ab.2015.12.017 (2016).
95. Feng, P. *et al.* iDNA6mA-PseKNC: Identifying DNA N 6-methyladenosine sites by incorporating nucleotide physicochemical properties into PseKNC. *Genomics* **111**(1), 96–102. https://doi.org/10.1016/j.ygeno.2018.01.005 (2019).
96. Hussain, W., Khan, Y. D., Rasool, N., Khan, S. A. & Chou, K. C. SPrenylC-PseAAC: A sequence-based model developed via Chou's 5-steps rule and general PseAAC for identifying S-prenylation sites in proteins. *J. Theor. Biol.* **468**, 1–11. https://doi.org/10.1016/j.jtbi.2019.02.007 (2019).
97. Ghauri, A. W., Khan, Y. D., Rasool, N., Khan, S. A. & Chou, K.-C. pNitro-Tyr-PseAAC: Predict nitrotyrosine sites in proteins by incorporating five features into Chou's general PseAAC. *Curr. Pharm. Des.* **24**(34), 4034–4043. https://doi.org/10.2174/1381612825666181127101039 (2018).
98. Khan, Y. D. *et al.* pSSbond-PseAAC: Prediction of disulfide bonding sites by integration of PseAAC and statistical moments. *J. Theor. Biol.* **463**, 47–55. https://doi.org/10.1016/j.jtbi.2018.12.015 (2019).
99. Khan, Y. D., Rasool, N., Hussain, W., Khan, S. A. & Chou, K. C. iPhosY-PseAAC: Identify phosphotyrosine sites by incorporating sequence statistical moments into PseAAC. *Mol. Biol. Rep.* **45**(6), 2501–2509. https://doi.org/10.1007/s11033-018-4417-z (2018).

Acknowledgements

The researchers would like to thank the Deanship of Scientific Research, Qassim University for funding the publication of this project.

Author contributions

All authors with equal collaboration, conceived and designed the experiments, performed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-19099-3.

Correspondence and requests for materials should be addressed to T.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.