EnhancerP-2L: A Gene regulatory site identification tool for DNA enhancer region using CREs motifs

^aAhmad Hassan Butt, ^bSalem Alkhalaf, ^cShaukat Iqbal, ^aYaser Daanial Khan,

^aDepartment of Computer Science, School of Systems and Technology, University of Management and Technology, Lahore, Pakistan

^bComputer Department, College of Arts and Sciences, Qassim University, AlRass, Saudi Arabia

^cDepartment of Informatics and Systems, School of Systems and Technology, University of Management and Technology, Lahore, Pakistan
{ahmad.hassan@umt.edu.pk, s.alkhalaf@qu.edu.sa, shaukat.iqbal@umt.edu.pk, yaser.khan@umt.edu.pk}

Abstract – Enhancers are DNA fragments that do not encode RNA molecules and proteins, but they act critically in the production of RNAs and proteins by controlling gene expression. Prediction of enhancers and their strength plays significant role in regulating gene expression. Prediction of enhancer regions, in sequences of DNA, is considered a difficult task due to the fact that they are not close to the target gene, have less common motifs and are mostly tissue/cell specific. In recent past, several bioinformatics tools were developed to discriminate enhancers from other regulatory elements and to identify their strengths as well. However the need for improvement in the quality of its prediction method requires enhancements in its application value practically. In this study, we proposed a new method that builds on nucleotide composition and statistical moment based features to distinguish between enhancers and non-enhancers and additionally determine their strength. Our proposed method achieved accuracy better than current state-of-the-art methods using 5-fold and 10-fold cross-validation. The outcomes from our proposed method suggest that the use of statistical moments based features could bear more efficient and effective results. For the accessibility of the scientific community, we have developed a user-friendly web server for EnhancerP-2L which will increase the impact of bioinformatics on medicinal chemistry and drive medical science into an unprecedented resolution. Web server is freely accessible at <http://www.biopred.org/enpred>.

Keywords: Gene expression; DNA enhancers; PRIM; Hahn Moments;

1. Introduction

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 (50-1500bp) segment of DNA and situated 1Mbp (1,000,000bp) 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].

About 30 years ago, a DNA segment, which was the first characterized enhancer, 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].

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 and resulted in high failure rates due to the fact that all enhancers do not have transcription factor occupations. The latter was to utilize DNase I hypersensitivity for enhancer predictions. Hence, leading to high false positive rate as many other DNA segments, which were non-enhancers, 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. 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.

2. Materials and Methods

2.1 Benchmark Dataset

The benchmark dataset of DNA enhancer sites, originally constructed and used in recent past [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 200bp fragments were extracted from DNA sequences. To remove pairwise sequences from the dataset, CD-HIT [30] tool was used to remove sequences having more than 20% similarity. The benchmark dataset includes 2,968 DNA enhancer sequences from which 1,484 are non-enhancer sequences and 1,484 are enhancer sequences. From 1,484 enhancer sequences, 742 are strong enhancers and 742 are weak enhancers for the second layer classification. Furthermore, the independent dataset used by a recent study [29] was utilized to enhance the effectiveness and performance of the proposed model. The independent dataset included 400 DNA enhancer sequences from which 200 (100 strong and 100 weak enhancers) are enhancers and 200 are non-enhancers. Table.1 and Table.2 includes the breakdown of the benchmark dataset.

In statistical based prediction models, the benchmark dataset mostly include training datasets and testing datasets. By utilizing various benchmark datasets, results obtained are computed from 5-fold and 10-fold cross-validations. The definition of a benchmark dataset is used in eqs.(1):

$$\begin{cases} D = D^+ \cup D^- \\ D^+ = D_{strong}^+ \cup D_{weak}^+ \end{cases} \quad (1)$$

Where D^+ contains 1,484 enhancers and D^- contains 1,484 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.

Table: 1 Breakdown of the benchmark datasets of DNA Enhancers and Non-Enhancers.

DNA Samples	Benchmark Dataset [26]	Independent Dataset [29]
Non-Enhancers	1,484	200
Enhancers	1,484	200
Overall	2,968	400

Table: 2 Breakdown of the DNA Enhancers from benchmark datasets.

DNA Enhancers	Benchmark Dataset [26]	Independent Dataset [29]
Strong Enhancers	742	100
Weak Enhancers	742	100
Total Enhancer	1,484	200

2.2 Feature Extraction

Today, one of the core requirements in the development of an effective bioinformatics predictor is the need 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 [31], 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) [32]. In almost all areas of bioinformatics and computational proteomics [31], 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' [33], 'PseAAC-Builder' [34] and 'PseAAC-General' [35] 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 [36] and the other two softwares generate Chou's special PseAAC in various modes [37]. 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 [36], "Functional Domain" mode [36] and "Sequential Evolution" mode or "PSSM" mode [36]. 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) [38] for generating different feature vectors for DNA/RNA sequences [39], [40] which proved very effective and efficient as well. In recent times a useful, efficient and a very powerful web server called 'Pse-in-One' [41] and its recently updated version 'Pse-in-One2.0' [42] were developed that are able to generate any preferred feature vector of pseudo components for DNA/RNA and protein/peptide sequences according to the requirements of study by any researcher.

In this study, we utilized the Kmer [43] 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 eqs.(2)

$$\mathbf{S} = Y_1 Y_2 Y_3 \dots Y_v \dots Y_w \quad (2)$$

Where Y_1 is represented as the first nucleotide of the DNA sample $\mathbf{S}$, Y_2 as the second nucleotide having the 2nd position of occurrence in DNA sample $\mathbf{S}$ 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 eqs.

(3)

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

Here $\in$ is the symbol used to represent the set theory ‘member of’ property and $1 \leq v \leq n$. 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.

2.2.1 Statistical Moments

Statistical moments are certain types of 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 [44]. 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 [45]–[48]. Several moments have been formed by various mathematicians and statisticians based on famous distribution functions and polynomials. To explicate the current problem, Raw; Central and Hahn moments are utilized [49].

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 [50]–[53]. 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. 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 eqs. (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 eqs. (4):

$$d = \lceil \sqrt{z} \rceil \quad (4)$$

Where the sample sequence length is ‘z’ and the 2-dimensional square matrix has ‘d’ as its dimension. The ordering obtained from equation (4) is used to form matrix M (eqs.(5)) having ‘m’ rows and ‘m’ columns.

$$M = \begin{bmatrix} N_{1 \rightarrow 1} & N_{1 \rightarrow 2} & \cdots & N_{1 \rightarrow j} & \cdots & N_{1 \rightarrow m} \\ N_{2 \rightarrow 1} & N_{2 \rightarrow 2} & \cdots & N_{2 \rightarrow j} & \cdots & N_{2 \rightarrow m} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ N_{k \rightarrow 1} & N_{k \rightarrow 2} & \cdots & N_{k \rightarrow j} & \cdots & N_{k \rightarrow m} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ N_{m \rightarrow 1} & N_{m \rightarrow 2} & \cdots & N_{m \rightarrow j} & \cdots & N_{m \rightarrow m} \end{bmatrix} \quad (5)$$

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

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

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

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 eqs.(7):

$$A_{jk} = \sum_a \sum_b a^j b^k f(a, b) \quad (7)$$

The raw moments of 2D matrix M, with order $(j + k)$ and up to a degree of 3, are computed using the eqs(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 $M_{00}, M_{01}, M_{10}, M_{11}, M_{02}, M_{20}, M_{12}, M_{21}, M_{30}$ & 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 [54], [55]. The central moments are also computed up to degree-3, using the centroid of the data as their reference point, from the following eqs.(8):

$$\mu_{jk} = \sum_a \sum_b (a - \bar{a})^j (b - \bar{b})^k f(a, b) \quad (8)$$

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

$$\bar{a} = \frac{M_{10}}{M_{00}}, \quad (9)$$

$$\bar{b} = \frac{M_{01}}{M_{00}} \quad (10)$$

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 eqs.(11):

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

The aforementioned Pochhammer symbol (P) was defined as follows in eqs.(12):

$$(\mathbb{P})_k = \mathbb{P}(\mathbb{P} + 1) \dots (\mathbb{P} + k - 1) \quad (12)$$

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

$$(\mathbb{P})_k = \frac{\Gamma(\mathbb{P} + k)}{\Gamma(\mathbb{P})} \quad (13)$$

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

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

Meanwhile, in eqs.(15),

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

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

$$H_{uv} = \sum_{b=0}^{N-1} \sum_{a=0}^{N-1} \beta_{ab} \widetilde{h}_u^{x,y}(b, N) \widetilde{h}_v^{x,y}(a, N), m, n = 0, 1, \dots, N - 1 \quad (16)$$

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

2.2.2 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 x 4) order. The matrix in eqs.(17) is used to extract position-relative attributes of every nucleotide in the given DNA sequence.

$$S_{D-PRIM} = \begin{bmatrix} N_{1 \rightarrow 1} & N_{2 \rightarrow 1} N_{3 \rightarrow 1} & N_{4 \rightarrow 1} \\ N_{1 \rightarrow 2} & N_{2 \rightarrow 2} N_{3 \rightarrow 2} & N_{4 \rightarrow 2} \\ N_{1 \rightarrow 3} & N_{2 \rightarrow 3} N_{3 \rightarrow 3} & N_{4 \rightarrow 3} \\ N_{1 \rightarrow 4} & N_{2 \rightarrow 4} N_{3 \rightarrow 4} & N_{4 \rightarrow 4} \end{bmatrix} \quad (17)$$

Here the indication score of the y^{th} position nucleotide is determined using $N_{x \rightarrow y}$ with respect to the x^{th} 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. The S_{D-PRIM} matrix is formed with 16 coefficient values obtained after successfully performing computations on position relative incidences.

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

2.2.3 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 effected 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. 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 x 4) order having 16 unique coefficients. It is defined by eqs.(18):

$$S_{D-RPRIM} = \begin{bmatrix} N_{1 \rightarrow 1} & N_{2 \rightarrow 1}N_{3 \rightarrow 1} & N_{4 \rightarrow 1} \\ N_{1 \rightarrow 2} & N_{2 \rightarrow 3}N_{3 \rightarrow 2} & N_{4 \rightarrow 2} \\ N_{1 \rightarrow 3} & N_{2 \rightarrow 3}N_{3 \rightarrow 3} & N_{4 \rightarrow 3} \\ N_{1 \rightarrow 4} & N_{2 \rightarrow 4}N_{3 \rightarrow 4} & N_{4 \rightarrow 4} \end{bmatrix} \quad (18)$$

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

2.2.4 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 eqs.(19):

$$\alpha = \{\mu_i, \dots, \mu_4\} \quad (19)$$

Here μ_i is the frequency of occurrence of the i^{th} ($1 \leq i \leq 4$) relevant nucleotide. Furthermore, the relative positions of nucleotides in any sequence is alleviated using these measures. The miscellany Super Feature Vector (SFV) includes these 4 features from FDV as unique attributes.

2.2.6 D-AAPIV (DNA-Accumulative-Absolute-Position-Incidence-Vector)

The distributional information of nucleotides was stored using frequency distribution vector which used the obscure features of DNA sequences 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 4 critical features associated with 4 native nucleotides in a DNA sequence. These 4 critical features from D-AAPIV are also added into the miscellany Super Feature Vector (SFV).

$$D-AAPIV = \{\mu_i, \dots, \mu_4\} \quad (20)$$

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

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

2.2.7 D-RAPIV (DNA-Reverse-Accumulative-Absolute-Position-Incidence-Vector)

D-RAPIV 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-RAPIV is formed as the following eqs.(24) using the reversed DNA sequence

and generates 4 valuable features. These 4 critical features from D-RAAPIV are also added into the miscellany Super Feature Vector (SFV).

$$D-RAAPIV = \{\alpha_1, \dots, \alpha_4\} \quad (22)$$

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

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

After calculating all possible features from the aforementioned extraction methods, the Super Feature Vector (SFV) of 134D features was obtained, for further processing in classification algorithm.

2.3 Classification Algorithm

2.3.1 Random Forests

In the past, ensemble learning methods have been applied in many bioinformatics relevant research studies [56], [57] 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 [58], [59] and bagging [60] 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.

Leo Breiman [61] built random forests, through which an additional layer of randomness is added to bagging. 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 (Figure 1 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 [56].

Fig.1: The structure of the Random Forest classifier

2.3.1.1 Algorithm: Supervised learning using Random Forest

Scikit-Learn [62] 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 25. According to [63], 128 is the theoretical upper limit of the trees and any further increase in the number of trees will not contribute to improve accuracy. One of the key findings observed during the experimentation process was that forest with more than 25 trees minimally contribute to the accuracy of the classifier, but can enhance the overall size of the proposed model substantially. Figure 2 illustrates a flowchart to show the overall process of the proposed method.

Fig.2: The Flowchart of the overall proposed method

3. 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.

3.1 Cross-validation

3.1.1 *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 $k = 1, 2, \dots, K$.

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, and the total number of observations to test is n in the dataset. The 5-fold and 10-fold 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 & 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 Table 5 and Table 6.

3.2 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 eqs (24):

$$\left\{ \begin{array}{l} Sn = \frac{TP}{TP + FN} \\ Sp = \frac{TN}{TN + FP} \\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \\ MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \end{array} \right. \quad (24)$$

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

eqs(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 [64] and Feng [65] using Chou's four intuitive equations which used the symbols introduced by Chou [66]. The symbols that define these equations are; Y^+ , Y^- , Y_+^+ and Y_+^- . The description of these symbols is defined in Table 3.

Table: 3 Description of symbols used to define these equations.

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

From the above correspondence in Table 4, we can define eqs.(25):

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

From the above correspondence in Table 4, we can define eqs.(26):

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

The above eqs.(26) has the same meaning as the equation (24) but it is more easy to understand and intuitive. Table 4 defines the detail description of these equations.

Table: 4 Description of equations used eqs. (26)

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.

The set of metrics used in above Table 4 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 [67]–[69]. The comparison of existing classifiers with proposed method is mentioned in Table.6.

3.3 Results and Discussions

The classification algorithms with their predictions results using benchmark dataset are shown in Table 5, 6 & 7. iEnhancer-EL [70] 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 [71] and iEnhancer-5Step [29] improved the accuracy results with their prediction model and acquired 77.86% 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 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 5 and Figure.3 for ROC). Furthermore, 10-fold cross-validation test was also conducted using random forest classifier on benchmark dataset and obtained the accuracy results are listed in Table 6. The ROCs 10-fold cross-validation tests are shown in Figure.4, 5, 6, 7, 8 and Figure.9. 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 7. Our proposed method is based on the feature sets that are evaluated using Hahn moments which are easier for the random forest based classifier 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.

Table: 5 Comparison of state-of-the-art methods with the proposed method using 5-Fold Cross validation tests

Layer	Classifiers	Sn(%)	Sp(%)	ACC(%)	MCC
1	iEnhancer-2L	78.09	75.88	76.89	0.54
	iEnhancer-2L-Hybrid	75.33	80.39	77.86	0.558
	EnhancerPred	72.57	73.79	80.82	0.464
	iEnhancer-EL	75.67	80.39	78.03	0.561
	iEnhancer-5Step	81.1	83.5	82.3	0.65
	Proposed Method	90.77	92.59	91.68	0.834
2	iEnhancer-2L	62.21	61.82	61.93	0.24
	iEnhancer-2L-Hybrid	71.02	60.64	65.83	0.318
	EnhancerPred	62.67	61.46	62.06	0.2413
	iEnhancer-EL	69	61.05	65.03	0.315
	iEnhancer-5Step	75.3	60.8	68.1	0.37
	Proposed Method	76.04	93.01	84.53	0.6975

Fig.3(a): ROC Curve of 5-fold cross validation tests for enhancers

Fig.3(b): ROC Curve of 5-fold cross validation tests for enhancer strengths

Table: 6 Comparison of classifiers for predicting enhancers using 10-fold cross validations

Layer	Classifier	Sn(%)	Sp(%)	ACC(%)	MCC
1	KNN	69.81	72.90	71.36	0.4275
	Naïve Bayes	67.59	69.47	68.53	0.3712
	AdaBoost	72.30	73.31	72.80	0.4569
	SVM	70.68	78.43	74.56	0.4933
	PNN	72.04	72.97	72.51	0.4507
	Random Forest	86.53	96.90	91.72	0.8398
2	KNN	58.77	54.05	56.41	0.1285
	Naïve Bayes	58.35	59.56	58.95	0.1793
	AdaBoost	63.46	57.15	60.31	0.2079
	SVM	69.94	55.68	62.80	0.2598
	PNN	76.95	44.34	60.64	0.2261
	Random Forest	80.49	93.97	87.23	0.7519

Fig.4(a): ROC Curve of 10-fold cross validation tests for enhancers using (KNN)

Fig.4(b): ROC Curve of 10-fold cross validation tests for enhancers strength (KNN)

Fig.5(a): ROC Curve of 10-fold cross validation tests for enhancers using (Naïve-Bayes)

Fig.5(b): ROC Curve of 10-fold cross validation tests for enhancers strength (Naïve-Bayes)

Fig.6(a): ROC Curve of 10-fold cross validation tests for enhancers using (AdaBoost)

Fig.6(b): ROC Curve of 10-fold cross validation tests for enhancers strength (AdaBoost)

Fig.7(a): ROC Curve of 10-fold cross validation tests for enhancers using (SVM)

Fig.7(b): ROC Curve of 10-fold cross validation tests for enhancers strength (SVM)

Fig.8(a): ROC Curve of 10-fold cross validation tests for enhancers using (PNN)

Fig.8(b): ROC Curve of 10-fold cross validation tests for enhancers strength (PNN)

Fig.9(a): ROC Curve of 10-fold cross validation tests for enhancers using (Random Forest)

Fig.9(b): ROC Curve of 10-fold cross validation tests for enhancers strength (Random Forest)

Table: 7 Independent tests based comparison of state-of-the-art methods with the proposed method

Layer	Classifiers	Sn(%)	Sp(%)	ACC(%)	MCC
1	iEnhancer-2L	71	75	73	0.46
	EnhancerPred	73.5	74.5	74	0.48
	iEnhancer-EL	71	78.5	74.75	0.496
	iEnhancer-5Step	82	76	79	0.58
	Proposed Method	78.10	81.05	79.50	0.5907
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
	Proposed Method	68.29	79.22	72.5	0.4624

3.4 Web-Server

As observed in past studies [72]–[76], 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 [77]–[81] to provide ease to biologists and scientists in drug discovery. The webserver which has been developed for the proposed method is freely accessible at <http://www.biopred.org/enpred> which is developed using Flask 1.1.1. The step-wise instructions to utilize the system are mentioned as follows:

Fig.10: The GUI of www.biopred.org/enpred webserver for enhancer predictions

3.4.1 Step-1

Open your web browser and navigate to www.biopred.org/enpred. The first page (see Figure.10) of the webserver is **Home**, page from where you can proceed to **ReadMe**, **Server**, **Data** and **Citations** pages through provided navigation links. The **Server** tab navigates to the portal for prediction process. **Data** tab facilitates the user with links to download the benchmark datasets used during the training and testing process. Finally, the **Citations** tab redirects the user to the page where information about the relevant paper and its citation is provided. In order to perform the prediction, kindly click on the **Server** tab.

3.4.2 Step-2

On the **Server** page, the user is provided with an empty text-area where the user can input the DNA Enhancer or DNA non-Enhancer sequence for prediction. The sequence input to the webserver is required to be in FASTA format. The **Submit** button will proceed to the prediction process for the input sequence. The results of the prediction process will appear on the **Results** page. The time of the prediction process totally depends upon the length of the input sequence.

3.4.3 Step-3

On the **Data** page, the user is provided with links to download the benchmark datasets for future experimentations.

3.4.4 Step-4

On the **ReadMe** page, the user is provided with relevant information about the current model which includes the details about the operation algorithm used for predictions.

4. Conclusions

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 of 91.68% and 84.53% for enhancer and strong enhancer classifications on a benchmark dataset which is currently the highest and accurate classification method for prediction of enhancers and their strength.

5. Compliance with ethical standards

5.1 Conflict of interest

All authors have declared that they have no conflict of interest.

5.2 Ethical approval

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

References

- [1] G. D. Erwin *et al.*, “Integrating Diverse Datasets Improves Developmental Enhancer Prediction,” *PLoS Comput. Biol.*, vol. 10, no. 6, pp. e1003677–e1003677, 2014.
- [2] A. Visel, E. M. Rubin, and L. A. Pennacchio, “Genomic views of distant-acting enhancers,” *Nature*, vol. 461, no. 7261, pp. 199–205, Sep. 2009.
- [3] N. J. Sakabe, D. Savic, and M. A. Nobrega, “Transcriptional enhancers in development and disease,” *Genome Biol.*, vol. 13, no. 1, p. 238, 2012.
- [4] N. D. Heintzman and B. Ren, “Finding distal regulatory elements in the human genome,” *Curr. Opin. Genet. Dev.*, vol. 19, no. 6, pp. 541–549, 2009.
- [5] E. M. Blackwood and J. T. Kadonaga, “Going the Distance: A Current View of Enhancer Action,” *Science (80-.)*, vol. 281, no. 60., 1998.
- [6] L. A. Pennacchio, W. Bickmore, A. Dean, M. A. Nobrega, and G. Bejerano, “Enhancers: five essential questions,” *Nat. Rev. Genet.*, vol. 14, no. 288., 2013.
- [7] O. I. Kulaeva, E. V Nizovtseva, Y. S. Polikanov, S. V Ulianov, and V. M. Studitsky, “Distant Activation of Transcription: Mechanisms of Enhancer Action,” *Mol. Cell. Biol.*, vol. 32, no. 24, pp. 4892–4897, 2012.
- [8] H.-M. Herz, “Enhancer deregulation in cancer and other diseases,” *BioEssays*, vol. 38, no. 10, pp. 1003–1015, 2016.
- [9] G. Zhang *et al.*, “DiseaseEnhancer: a resource of human disease-associated enhancer catalog,” *Nucleic Acids*

- Res.*, vol. 46, no. D1, pp. D78–D84, 2017.
- [10] O. Corradin and P. C. Scacheri, “Enhancer variants: evaluating functions in common disease,” *Genome Med.*, vol. 6, no. 85., 2014.
 - [11] M. Boyd *et al.*, “Characterization of the enhancer and promoter landscape of inflammatory bowel disease from human colon biopsies,” *Nat. Commun.*, vol. 9, no. 1661., 2018.
 - [12] J. Banerji, S. Rusconi, and W. Schaffner, “Expression of a β -globin gene is enhanced by remote SV40 DNA sequences,” *Cell*, vol. 27, no. 2, pp. 299–308, 1981.
 - [13] D. Shlyueva, G. Stampfel, and A. Stark, “Transcriptional enhancers: From properties to genome-wide predictions,” *Nat. Rev. Genet.*, vol. 15, no. 4, pp. 272–286, 2014.
 - [14] N. D. Heintzman *et al.*, “Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome,” *Nat. Genet.*, vol. 39, no. 3, p. 311, 2007.
 - [15] F. Jin, Y. Li, B. Ren, and R. Natarajan, “PU. 1 and C/EBP α synergistically program distinct response to NF- κ B activation through establishing monocyte specific enhancers,” *Proc. Natl. Acad. Sci.*, vol. 108, no. 13, pp. 5290–5295, 2011.
 - [16] T.-K. Kim *et al.*, “Widespread transcription at neuronal activity-regulated enhancers,” *Nature*, vol. 465, no. 7295, p. 182, 2010.
 - [17] A. P. Boyle *et al.*, “High-resolution genome-wide in vivo footprinting of diverse transcription factors in human cells,” *Genome Res.*, vol. 21, no. 3, pp. 456–464, 2011.
 - [18] A. Visel *et al.*, “ChIP-seq accurately predicts tissue-specific activity of enhancers,” *Nature*, vol. 457, no. 7231, pp. 854–858, 2009.
 - [19] J. Ernst *et al.*, “Mapping and analysis of chromatin state dynamics in nine human cell types,” *Nature*, vol. 473, no. 7345, pp. 43–49, 2011.
 - [20] M. Fernández and D. Miranda-Saavedra, “Genome-wide enhancer prediction from epigenetic signatures using genetic algorithm-optimized support vector machines,” *Nucleic Acids Res.*, vol. 40, no. 10, pp. e77–e77, 2012.
 - [21] H. A. Firpi, D. Ucar, and K. Tan, “Discover regulatory DNA elements using chromatin signatures and artificial neural network,” *Bioinformatics*, vol. 26, no. 13, pp. 1579–1586, 2010.
 - [22] D. Klefogiannis, P. Kalnis, and V. B. Bajic, “DEEP: A general computational framework for predicting enhancers,” *Nucleic Acids Res.*, vol. 43, no. 1, p. e6, 2015.
 - [23] N. Rajagopal *et al.*, “RFECS: A Random-Forest Based Algorithm for Enhancer Identification from Chromatin State,” *PLoS Comput. Biol.*, vol. 9, no. 3, pp. e1002968–e1002968, 2013.
 - [24] H. Bu, Y. Gan, Y. Wang, S. Zhou, and J. Guan, “A new method for enhancer prediction based on deep belief network,” *BMC Bioinformatics*, vol. 18, no. 418., 2017.
 - [25] B. Yang *et al.*, “BiRen: predicting enhancers with a deep-learning-based model using the DNA sequence alone,” *Bioinformatics*, vol. 33, no. 13, pp. 1930–1936, 2017.
 - [26] B. Liu, L. Fang, R. Long, X. Lan, and K. C. Chou, “iEnhancer-2L: A two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition,” *Bioinformatics*, vol. 32, no. 3, pp.

- 362–369, 2016.
- [27] C. Jia and W. He, “EnhancerPred: A predictor for discovering enhancers based on the combination and selection of multiple features,” *Sci. Rep.*, vol. 6, 2016.
- [28] W. He and C. Jia, “EnhancerPred2.0: Predicting enhancers and their strength based on position-specific trinucleotide propensity and electron-ion interaction potential feature selection,” *Mol. Biosyst.*, vol. 13, no. 4, pp. 767–774, 2017.
- [29] N. Q. K. Le, E. K. Y. Yapp, Q. T. Ho, N. Nagasundaram, Y. Y. Ou, and H. Y. Yeh, “iEnhancer-5Step: Identifying enhancers using hidden information of DNA sequences via Chou’s 5-step rule and word embedding,” *Anal. Biochem.*, vol. 571, pp. 53–61, 2019.
- [30] L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li, “CD-HIT: Accelerated for clustering the next-generation sequencing data,” *Bioinformatics*, vol. 28, no. 23, pp. 3150–3152, 2012.
- [31] K.-C. Chou, “Impacts of Bioinformatics to Medicinal Chemistry,” *Med. Chem. (Los. Angeles).*, vol. 11, no. 3, pp. 218–234, 2015.
- [32] K. C. Chou, “Prediction of protein cellular attributes using pseudo-amino acid composition,” *Proteins Struct. Funct. Genet.*, vol. 43, no. 3, pp. 246–255, 2001.
- [33] D.-S. Cao, Q.-S. Xu, and Y.-Z. Liang, “propy: a tool to generate various modes of Chou’s PseAAC,” *Bioinformatics*, vol. 29, no. 7, pp. 960–962, 2013.
- [34] P. Du, X. Wang, C. Xu, and Y. Gao, “PseAAC-Builder: a cross-platform stand-alone program for generating various special Chou’s pseudo-amino acid compositions,” *Anal. Biochem.*, vol. 425, pp. 117–119, 2012.
- [35] P. Du, S. Gu, and Y. Jiao, “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.*, vol. 15, no. 3495, 2014.
- [36] K. C. Chou, “Some remarks on protein attribute prediction and pseudo amino acid composition,” *J. Theor. Biol.*, vol. 273, no. 1, pp. 236–247, 2011.
- [37] K.-C. Chou, “Pseudo Amino Acid Composition and its Applications in Bioinformatics, Proteomics and System Biology,” *Curr. Proteomics*, vol. 6, no. 4, pp. 262–274, 2009.
- [38] W. Chen, T. Y. Lei, D. C. Jin, H. Lin, and K. C. Chou, “PseKNC: A flexible web server for generating pseudo K-tuple nucleotide composition,” *Anal. Biochem.*, vol. 456, no. 1, pp. 53–60, 2014.
- [39] W. Chen, H. Lin, and K. C. Chou, “Pseudo nucleotide composition or PseKNC: An effective formulation for analyzing genomic sequences,” *Mol. Biosyst.*, vol. 11, no. 10, pp. 2620–2634, 2015.
- [40] B. Liu, F. Yang, D.-S. Huang, and K.-C. Chou, “iPromoter-2L: a two-layer predictor for identifying promoters and their types by multi-window-based PseKNC,” *Bioinformatics*, vol. 34, no. 1, pp. 33–40, 2017.
- [41] B. Liu, F. Liu, X. Wang, J. Chen, L. Fang, and K. C. Chou, “Pse-in-One: A web server for generating various modes of pseudo components of DNA, RNA, and protein sequences,” *Nucleic Acids Res.*, vol. 43, no. W1, pp. W65–W71, 2015.
- [42] B. Liu, H. Wu, and K.-C. Chou, “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.*, vol. 09, no. 04, pp. 67–91, 2017.
- [43] B. Liu, R. Long, and K. C. Chou, “IDHS-EL: Identifying DNase i hypersensitive sites by fusing three different modes of pseudo nucleotide composition into an ensemble learning framework,” *Bioinformatics*, vol. 32, no. 16, pp. 2411–2418, 2016.
- [44] R. C. Papademetriou, “Reconstructing with moments,” *Proc. - Int. Conf. Pattern Recognit.*, vol. 3, pp. 476–480, 1992.
- [45] A. H. Butt, S. A. Khan, H. Jamil, N. Rasool, and Y. D. Khan, “A Prediction Model for Membrane Proteins Using Moments Based Features,” *Biomed Res. Int.*, vol. 2016, pp. 1–7, 2016.
- [46] A. H. Butt, N. Rasool, and Y. D. Khan, “A Treatise to Computational Approaches Towards Prediction of Membrane Protein and Its Subtypes,” *J. Membr. Biol.*, vol. 250, no. 1, pp. 55–76, 2017.
- [47] A. H. Butt, N. Rasool, and Y. D. Khan, “Predicting membrane proteins and their types by extracting various sequence features into Chou’s general PseAAC,” *Mol. Biol. Rep.*, vol. 45, no. 6, pp. 2295–2306, 2018.
- [48] A. H. Butt, N. Rasool, and Y. D. Khan, “Prediction of antioxidant proteins by incorporating statistical moments based features into Chou’s PseAAC,” *J. Theor. Biol.*, vol. 473, pp. 1–8, 2019.
- [49] Y. D. Khan, S. A. Khan, F. Ahmad, and S. Islam, “Iris recognition using image moments and k-Means algorithm,” *Sci. World J.*, vol. 2014, pp. 1–9, 2014.
- [50] J. Zhou, H. Shu, H. Zhu, C. Toumoulin, and L. Luo, “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 LNCS, 2005, pp. 524–531.
- [51] H. Zhu, H. Shu, J. Zhou, L. Luo, and J. L. Coatrieux, “Image analysis by discrete orthogonal dual Hahn moments,” *Pattern Recognit. Lett.*, vol. 28, no. 13, pp. 1688–1704, 2007.
- [52] P. T. Yap, R. Paramesran, and S. H. Ong, “Image analysis using Hahn moments,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11, pp. 2057–2062, 2007.
- [53] H.-A. GOH, C.-W. CHONG, R. BESAR, F. S. ABAS, and K.-S. SIM, “Translation and Scale Invariants of Hahn Moments,” *Int. J. Image Graph.*, vol. 09, no. 02, pp. 271–285, 2009.
- [54] M. A. Akmal, N. Rasool, and Y. D. Khan, “Prediction of N-linked glycosylation sites using position relative features and statistical moments,” *PLoS One*, vol. 12, no. 8, pp. e0181966–e0181966, 2017.
- [55] Y. D. Khan, A. Batool, N. Rasool, S. A. Khan, and K.-C. Chou, “Prediction of Nitrosocysteine Sites Using Position and Composition Variant Features,” *Lett. Org. Chem.*, vol. 16, no. 4, pp. 283–293, 2018.
- [56] A. Tyryshkina, N. Coraor, and A. Nekrutenko, “Predicting runtimes of bioinformatics tools based on historical data: five years of Galaxy usage,” *Bioinformatics*, vol. 35, no. 18, pp. 3453–3460, 2019.
- [57] N. Simidjievski, L. Todorovski, and S. Džeroski, “Modeling dynamic systems with efficient ensembles of process-based models,” *PLoS One*, vol. 11, no. 4, 2016.
- [58] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 904, no. 1, pp. 23–37, Aug. 1995.

- [59] R. E. Schapire, “Theoretical, views of boosting and applications,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1720, 1999, pp. 13–25.
- [60] L. Breiman, “Bagging predictors,” *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
- [61] L. Breiman, “Random Forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [62] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Jan. 2011.
- [63] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, “How many trees in a random forest?,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7376 LNAI, 2012, pp. 154–168.
- [64] Y. Xu, X. J. Shao, L. Y. Wu, N. Y. Deng, and K. C. Chou, “ISNO-AAPair: Incorporating amino acid pairwise coupling into PseAAC for predicting cysteine S-nitrosylation sites in proteins,” *PeerJ*, vol. 2013, no. 1, pp. e171–e171, 2013.
- [65] P. M. Feng, H. Ding, W. Chen, and H. Lin, “Naïve bayes classifier with feature selection to identify phage virion proteins,” *Comput. Math. Methods Med.*, vol. 2013, pp. 1–6, 2013.
- [66] K. C. Chou, “Prediction of signal peptides using scaled window.,” *Peptides*, vol. 22, no. 12, pp. 1973–1979, 2001.
- [67] X. Xiao, P. Wang, W. Z. Lin, J. H. Jia, and K. C. Chou, “IAMP-2L: A two-level multi-label classifier for identifying antimicrobial peptides and their functional types,” *Anal. Biochem.*, vol. 436, no. 2, pp. 168–177, 2013.
- [68] X. Xiao, Z. C. Wu, and K. C. Chou, “iLoc-Virus: A multi-label learning classifier for identifying the subcellular localization of virus proteins with both single and multiple sites,” *J. Theor. Biol.*, vol. 284, no. 1, pp. 42–51, 2011.
- [69] W. Z. Lin, J. A. Fang, X. Xiao, and K. C. Chou, “iLoc-Animal: A multi-label learning classifier for predicting subcellular localization of animal proteins,” *Mol. Biosyst.*, vol. 9, no. 4, pp. 634–644, 2013.
- [70] B. Liu, K. Li, D. S. Huang, and K. C. Chou, “iEnhancer-EL: Identifying enhancers and their strength with ensemble learning approach,” *Bioinformatics*, vol. 34, no. 22, pp. 3835–3842, 2018.
- [71] M. Tahir, M. Hayat, and S. A. Khan, “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.*, vol. 43, no. 12, pp. 6719–6727, 2018.
- [72] X. Cheng, X. Xiao, and K. C. Chou, “pLoc_bal-mGneg: Predict subcellular localization of Gram-negative bacterial proteins by quasi-balancing training dataset and general PseAAC,” *J. Theor. Biol.*, vol. 458, pp. 92–102, 2018.
- [73] K.-C. Chou, “Proposing Pseudo Amino Acid Components is an Important Milestone for Proteome and Genome Analyses,” *Int. J. Pept. Res. Ther.*, 2019.
- [74] B. Liu, H. Wu, D. Zhang, X. Wang, and K. C. Chou, “Pse-Analysis: A python package for DNA/RNA and protein/ peptide sequence analysis based on pseudo components and kernel methods,” *Oncotarget*, vol. 8,

- no. 8, pp. 13338–13343, 2017.
- [75] Z. Liu, X. Xiao, D. J. Yu, J. Jia, W. R. Qiu, and K. C. Chou, “pRNA_m-PC: Predicting N⁶-methyladenosine sites in RNA sequences via physical-chemical properties,” *Anal. Biochem.*, vol. 497, pp. 60–67, 2016.
- [76] P. Feng, H. Yang, H. Ding, H. Lin, W. Chen, and K. C. Chou, “iDNA_{6m}A-PseKNC: Identifying DNA N⁶-methyladenosine sites by incorporating nucleotide physicochemical properties into PseKNC,” *Genomics*, vol. 111, no. 1, pp. 96–102, 2019.
- [77] W. Hussain, Y. D. Khan, N. Rasool, S. A. Khan, and K. C. Chou, “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.*, vol. 468, pp. 1–11, 2019.
- [78] A. W. Ghauri, Y. D. Khan, N. Rasool, S. A. Khan, and K.-C. Chou, “pNitro-Tyr-PseAAC: Predict Nitrotyrosine Sites in Proteins by Incorporating Five Features into Chou’s General PseAAC,” *Curr. Pharm. Des.*, vol. 24, no. 34, pp. 4034–4043, 2018.
- [79] Y. D. Khan, M. Jamil, W. Hussain, N. Rasool, S. A. Khan, and K. C. Chou, “pSSbond-PseAAC: Prediction of disulfide bonding sites by integration of PseAAC and statistical moments,” *J. Theor. Biol.*, vol. 463, pp. 47–55, 2019.
- [80] Y. D. Khan, N. Rasool, W. Hussain, S. A. Khan, and K. C. Chou, “iPhosY-PseAAC: identify phosphotyrosine sites by incorporating sequence statistical moments into PseAAC,” *Mol. Biol. Rep.*, vol. 45, no. 6, pp. 2501–2509, 2018.
- [81] M. Awais, W. Hussain, Y. D. Khan, N. Rasool, S. A. Khan, and K. C. Chou, “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. Bioinforma.*, p. 1, 2019.