Feature Selection Method for DNA-Binding Protein Identification

Author:
Shahana Yasmin Chowdhury
Student ID: 012171004

A Thesis in
the Department of
Computer Science and Engineering

Submitted in Partial Fulfillment of the Requirements
For the Degree of Master of Science in Computer Science and Engineering
United International University, Dhaka, Bangladesh

December 3, 2017

Declaration

This is to certify that the work entitled "Feature Selection Method for DNA-Binding Protein Identification" is the outcome of the research carried out by me under the supervision of Dr. Swakkhar Shatabda, Asst. Professor, Dept of CSE, United International University.

Shahana Yasmin Chowdhury
Student ID - 012171004, CSE
United International University

In my capacity as supervisor of the candidates thesis, I certify that the above statements are true to the best of my knowledge.

Dr. Swakkhar Shatabda
Asst. Professor, Dept of CSE
United International University.

Contents

List of Figures

List of Tables

Abstract

DNA-binding proteins play a very important role in the structural composition of the DNA. In addition to that they regulate and effect various cellular processes like transcription, DNA replication, DNA recombination, repair and modification. The experimental methods used to identify DNA-binding proteins are expensive and time consuming and thus attracted researchers from computational field to address the problem. In this paper, we present iDNAProt-ES, a DNA binding protein prediction method that utilizes both sequence based evolutionary and structure based features of proteins to identify their DNA-binding functionality. We used recursive feature elimination to extract an optimal set of features and train them using support vector machine with linear kernel to select the final model. Our proposed method significantly outperforms the existing state-of-the-art predictors on standard benchmark dataset. The accuracy of the predictor is 90.18% using jack knife test and 88.87% using 10-fold cross validation on the benchmark dataset. The accuracy of the predictor on the independent dataset is 80.64% which is also significantly improved than the state-of-the-art methods. iDNAProt-ES is a novel prediction method that uses evolutionary and structure based features. We believe the superior performance of iDNAProt-ES will motivate the researchers to use these set of features and the methods and also help the researchers interested in identification of DNA-binding proteins. iDNAProt-ES is readily available to be use as a web server from: http://brl.uiu.ac.bd/iDNAProt-ES/.

Submitted for Publications

1 Chowdhury. Shahana Yasmin, Swakkhar Shatabda, and Abdollah Dehzangi. "iDNAProt-ES: Identification of DNA-binding Proteins Using Evolutionary and Structural Features." Scientific Reports 7.1 (2017): 14938.

2 Shahana Yasmin chowdhury, Swakkhar Shatabda and Abdollah Dehzangi. "Greedy Step Wise Feature Selection for DNA-Binding Protein Prediction Problem".

Acknowledgments

Here I want to express my gratitude and sincere appreciation to all those, who have helped me complete this Master Thesis.

I want to especially thank to my supervisor Dr. Swakkhar Shatabda, Asst Professor, Dept of CSE, UIU, for his numerous corrections, advises and support, both during the actual work process and when writing the thesis. Without his guidance it's impossible for me to complete the thesis.

I am also grateful to Dr. Abdollah Dehzangi, Department of Computer Science, Morgan State University, Baltimore, Maryland, United States, for reviewing of the thesis and verification of the results.

My heartfelt gratitude and appreciation to the panel of Examiners Dr. M. Sohel Rahman, Head of the Dept., Dept of CSE, BUET, for his constructive comments suggestions and critiquing.

I would like to express my deepest thanks to Dr. Mohammad Nurul Huda, Professor, Dept of CSE, UIU, for his favorable response regarding the study.

Last but not least, I want to thank all the people who have willingly helped me out with their abilities.

Chapter 1

Introduction

DNA (Deoxyribonucleic Acid) binding proteins play a very important role in the structural composition of the DNA. In addition to that they regulate and effect various cellular processes like transcription, DNA replication, DNA recombination, repair and modification. The experimental methods used to identify DNA-binding proteins are expensive and time consuming and thus attracted researchers from computational field to address the problem.

1.1 Problem Description

DNA (Deoxyribonucleic Acid) binding proteins are those proteins that bind with single or double stranded DNA. DNA-binding proteins plays an important role in the structural composition of the DNA and in gene regulations. Non-specific structural proteins often help to organize and compact the chromosomal DNA. The other important role is to regulate and a effect various cellular processes like transcription, DNA replication, DNA recombination, repair and modification. These proteins in their independently folded domains have at least one structural motif that recognizes and has an affinity to DNA [13]. DNA binding proteins or ligands have many important applications as antibiotics, drugs, steroids for various biological effects and in bio-physical, bio-chemical and biological studies of DNA [14].

Many experimental methods are being used to identify DNA-binding proteins: filter binding assays [15], genetic analysis [16], X-ray crystallography [17], chromatin immunoprecipitation on microarrays [18], Nuclear Magnetic Resonance (NMR) [19] etc. Due to excessive cost of these techniques [20], many computational methods have been applied by the researchers to identify DNA-binding proteins. Moreover, the number of newly discovered protein sequences has been increasing extremely fast due to the advent of modern protein sequencing technologies. For example, in 1986 the Swiss-Prot [21] database contained only 3,939 protein sequence entries, but now the number has jumped to 554515 according to the release 2017_05 of May, 10, 2017 by the UniProtKB/Swiss-Prot [1]. It means that the number of protein sequence entries now is more than 140 times the number from about 25 years ago. Facing the flood of new protein sequences generated in the post genomic age, it is highly desirable to develop automated computational prediction approaches for rapidly and effectively identifying and characterizing DNA-binding proteins.

1.2 Motivation

The method of DNA-binding Protein has various functions in the cell and plays an important role in the structural composition of the DNA and in gene regulations. Non-specific structural proteins often help to organize and compact the chromosomal DNA. The other important role is to regulate and effect various cellular processes like transcription, DNA replication, DNA recombination, repair and modification. So our study and result would be notably helpful for many researchers in these areas. We expect that our contribution will speed up the future research.

1.3 Aims & Objectives

This research aims to establish a DNA-binding protein prediction method that utilizes both sequence based evolutionary and structural based features

[1] `http://web.expasy.org/docs/relnotes/relstat.html`

of proteins to identify their DNA-binding functionality. It is intended that the research findings will contribute to the development of a web server to predict new protein sequences.

These above aims raise the following core objectives:

- To find out and prepare a Benchmark data set.

- To extract adequate number of features from the protein evolutionary and structural information of protein.

- To design and run different machine learning classification algorithms.

- To analyze data and results to find the best algorithm for the Benchmark data set..

- To reduce the feature set to improve accuracy than the state-of-the-art.

- To develop a prediction web server based on our methodology to predict the protein.

1.4 Methodology

According to the methodology mentioned by Liu[6]we followed the below procedures to establish a predictor for DNA-binding Protein identification with a novel feature set.

- Construct or select a valid benchmark dataset to train and test the predictor.

- Formulate the protein samples with an effective mathematical expression that can truly reflect their intrinsic correlation with the attribute to be predicted.

- Introduce or develop a powerful algorithm (or engine) to operate the prediction.

- Perform properly cross-validation tests to objectively evaluate the anticipated accuracy of the predictor.

1.5 Contribution

Our contribution in this thesis is mentioned below.

At first we select two reliable datasets -one as the Benchmark and the other as the independent set. Then we prepared the feature set from both the datasets. After that we implement a feature reduction method to enhance prediction accuracy. Then we train and test the model by using the Support Vector Machine (SVM) and perform cross-validation tests to evaluate the accuracy of our predictor. Finally we developed a web-server to predict DNA-binding protein. This web-server is publicly available at http://brl.uiu.ac.bd/iDNAProt-ES/.

1.6 Organization

The organization of this document is described below. In this first chapter we discussed the problem description, motivation, aims & objectives of the thesis, methodology and our main contribution. In the next chapter we will discuss some terminologies and background of the DNA-binding protein. In the third chapter we will focus on materials and methods including datasets, feature selection method, algorithm, description of the classifier and performance evaluation process etc. In the fourth chapter we will present and discuss about all kinds of experiments and results thereof which we got in this study. In the fifth chapter we will conclude briefly identifying the limitation of this research and discussing possible future works.

Chapter 2

Background

This chapter describes the notions and notations on DNA, protein, structure of the protein, Central Dogma, DNA-binding protein and PSSM etc. And finally it will show some related work on DNA-binding proteins and their usefulness and limitation.

2.1 Biological Preliminaries

2.1.1 DNA

The full form of DNA is deoxyribonucleic acid. It is a molecule. The smallest unit of DNA is atom. Lots of atom join as a string and makes the long spiral shape of ladder to form DNA [22]. DNA are the blueprint of the living things. The chemical that made the DNA structure is called nucleotides. It has four types of nucleotides i.e., adenine, thymine, cytosine and guanine. Short form of nucleotides are A,T,C and G. To make the ladder A pairs with T and C pair with G. A-T base pairs connected with 2 hydrogen bonds and C-G base pair is connected with 3 hydrogen bonds. This type of structure is called the double stranded DNA. Figure 2.1 shows the DNA structure with major and minor grooves. When the DNA strands are close together called minor grooves, when the distance is high between the strands is called major

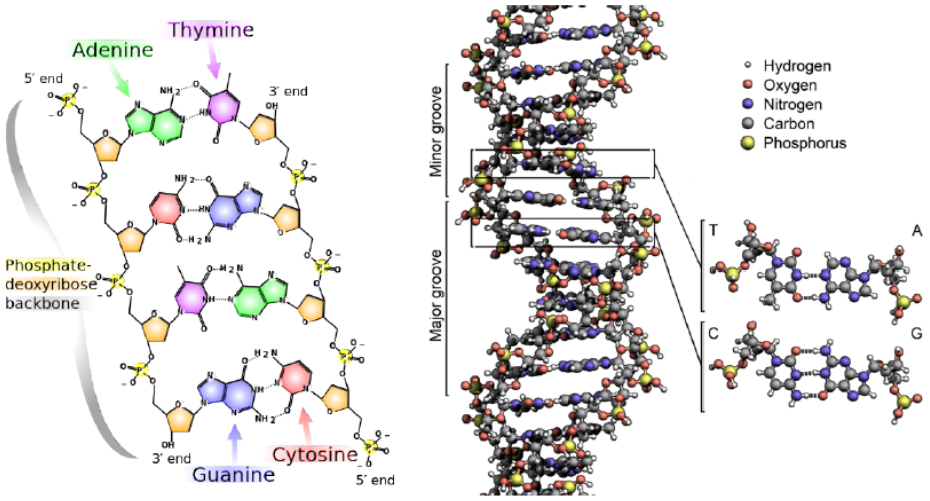

Figure 2.1: The DNA structure with major and minor groove. [1].

groove.The grooves are important for the DNA-protein interaction.

2.1.2 Protein

Proteins are the poly-peptide chains [23]. poly-peptide chains are made of a string of smaller building blocks called amino acids. Amino acids join together and form different structures. When we eat food we breaks down the proteins that we ate into amino acids and then we stick them back together again to make lots of different types of proteins to do the jobs in cell. For example, Proteins can do the jobs like, detect light, form hair, fight disease by antibody etc. The sequence of amino acids in any protein is decided by the sequence of genes which is encoded in the genetic code. We have different types of protein functionalities. Some example are listed below.

- Protein can moves us. The reason is that meat is muscle. It has contract to move our body.

- Protein can kill us. Botox (Botulinum Toxin) is a protein. Botox cuts up the molecules in nerve cells which control our muscles This means

the Botox makes our muscle relax. The muscles which control our lungs stop working after taking food contaminated with botulism.

- Protein speed things up. Enzymes are that makes reactions go more quickly. For example, those helps break down starch to sugar.

- Its transport oxygen in our body by red blood cell which contain protein compound called hemoglobin.

2.1.3 Protein Structure

To understand the protein structures we have to know about the basic structure of an amino acid. There is a carbolic acid group (-COOH) on one side and an amino group (-NH2) on the other side. They may have the side chains which are unique to each individual amino acid. There are 20 different standard amino acids used by cells for protein construction[2]. Biochemists often refer to four distinct aspects of a protein's structure. For levels of protein structures is shown in fig 2.2:

i. Primary structure: It's nothing but amino acids attached with each other.Its a linear chain of amino acid resedues in a polypeptide chain. Each polypeptide chain has a N-Terminus and C-Terminus. Bonding is done by covalent forces like peptide bonds.

ii. Secondary structure: There are hydrogen bonds forming between the residues of main-chain peptide groups. As a result, it gives structures Alpha Helix, Beta Sheet and Beta Turn [2]. Alpha Helices are the right-handed spiral strands. The Beta sheet consists of pairs of strands lying side-by-side.

iii. Tertiary structure: Two or more Alpha Helices and/or Beta sheets further fold to form the Tertiary Structure. The tertiary structures are formed by non-covalent bonds such as hydrogen bonding and Van-der Waals forces and covalent bonds such as disulfied bonds. Insulin is an example of this type of structure.

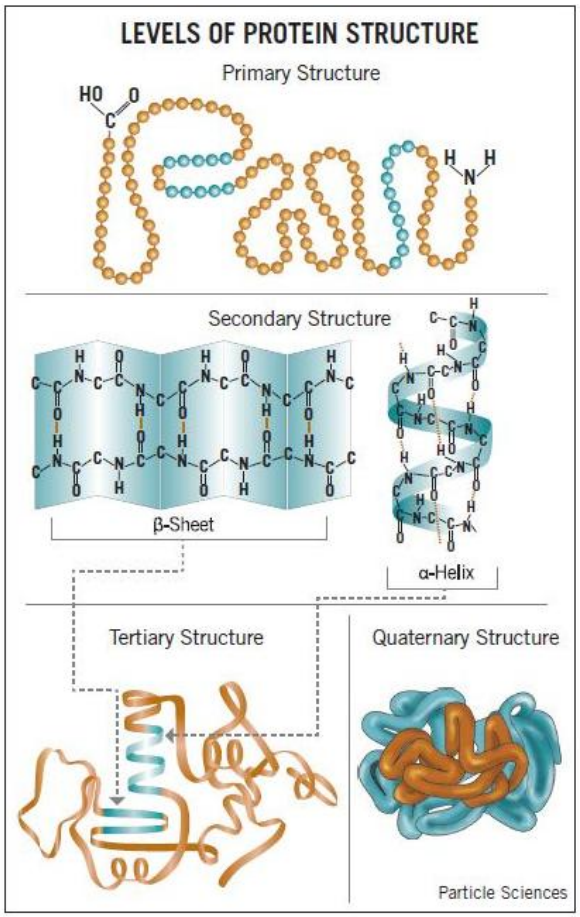

Figure 2.2: Lavels of Protein Structure [2].

iv. Quaternary structure: Two or more polypeptide units form the Quaternary structure by different interactions. All the polipeptide units must contain tertiary structures. Bonding is via non-covalent bonds such as hydrogen bondings, salt bridges etc. Hemoglobin is an example of a Quaternary structure.

2.1.4 Protein Synthesis

Cell is the basic unit of all living tissues. There is a structure called nucleus in the cell. The nucleus contains the genome, and it can be spitted in different pair of chromosomes based on the living things. Each chromosome contains the long strand of DNA, tightly packaged around the histones protein. DNA sections are called the gene or genetic code. Each gene contains the instruction for making proteins. When a gene is switched on an enzyme called RNA polymerase is attached to the gene. It moves along the DNA strand to make the messenger RNA (mRNA) out of free bases in the nucleus. In the DNA strand C-G nucleotides match up and A-T nucleotides match up. But free bases are added to mRNA and A matches up with U (Urasil) in place of T. This process of DNA portion to mRNA production is called transcription. After forming the mRNA it moves from the nucleus to cytoplasm.In this stage mRNA needs a ribosome, which is called the protein factory in the cytoplasm.Ribosomes are bound to the mRNA and reads the code in the mRNA to produce a chain made up of amino acids. There are 20 different types of amino acid. The Transfer RNA (tRNA) molecules carry the amino acids to the ribosome. Three bases (or codon) are read at a time from mRNA and delivers the corresponding amino acid. This is added to a growing chain of amino acids. The amino acid chain grows until a stop codon is reached. Once the stop codon is read, the chain folds into a complex three dimensional shape to form the protein. The process of mRNA to protein by tRNA is called translation. The whole process of transcription and translation to produce protein from DNA is called the protein synthesis, shown in Figure 2.3. In molecular biology, the precess of making proteins from DNA is also known as Central Doghma.

2.1.5 DNA-binding Protein

DNA-binding proteins are the proteins that are able bind to the single or double stranded DNA. Sequence specific DNA-binding proteins normally interact with the major groove of DNA [24]. Non-specific interaction occurs with DNA phosphate backbones. DNA-binding proteins contain the DNA-binding domain. The example of DNA-binding domains are zinc finger, helix-turn-helix and leucine zipper etc. These are the different types of structural

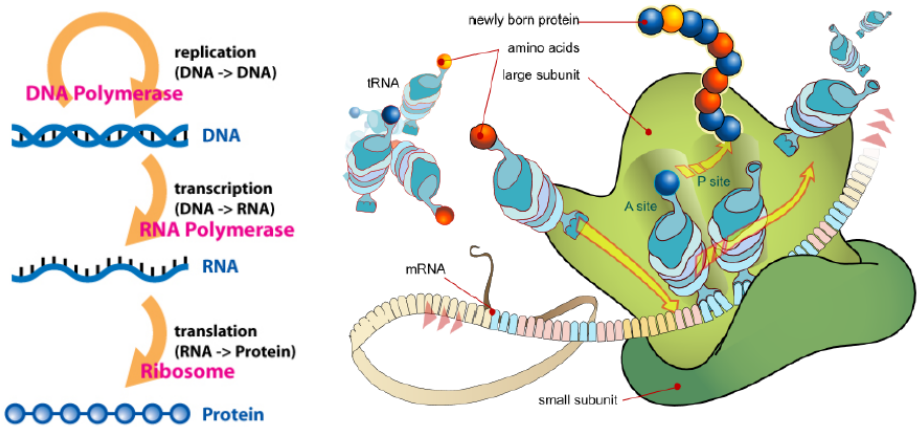

Figure 2.3: The Central Dogma of molecular biology (left): replication, transcription and translation [3]. Translation of mRNA (right) to a protein by the ribosome which consists of a small and large subunit [4].

domain in protein. Figure 2.4 shows a DNA-binding protein.

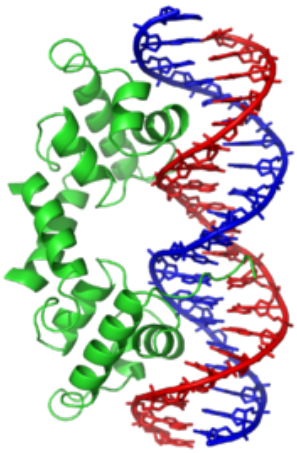

Figure 2.4: The lambda repressor helix-turn-helix transcription factor bound to its DNA target[5].

2.1.6 PSSM

Position-Specific Scoring Matrix (PSSM) used to represent the patterns of biological sequence. The matrices are derived from the set of aligned sequences. The PSSM's profiles are the matrices without gap information and without substitution table information of the alignment of the biological sequence. Smith-Waterman algorithm[25] is used to find the scoring matrix. PSSMs can be created using PSI-BLAST [26], which finds similar protein sequences to a query sequence, and then constructs a PSSM from the resulting alignment. PSSM scores are generally shown as positive or negative integers. Positive scores indicate that the given amino acid substitution occurs more frequently in the alignment than expected by chance, while negative scores indicate that the substitution occurs less frequently than expected.

2.2 Literature Review

Computational methods that have been used to predict the DNA-binding proteins can be broadly categorized into two groups: structure based methods [27, 11] and sequence based methods [28, 6, 29, 30]. In most of the cases, DNA-binding protein identification is formulated as a binary classification problem in the supervised learning setting. The sequence based methods are built depending only on the sequence based information extracted from the training data where structure based methods also exploits structure based features. In [12], structural motifs and electrostatic potentials were used to predict DNA-binding proteins. The iDBPs server was proposed in [31] that used global features like average surface electrostatic potential, the dipole moment and cluster-based amino acid conservation patterns. One of the major difficulties in structure based methods is that the structure of most of the proteins are unknown. However, structural information like presence of motifs and other information are very crucial in DNA recognition of binding proteins. Therefore, we hypothesize that even partial information of the protein structure could play very important role in identifying their functions of binding DNA.

Many machine learning algorithms are applied to solve this problem in the

literature. Among these are: Logistic Regression [10], Hidden Markov Models [12], Random Forest [31, 8, 7], Artificial Neural Network [32], Support Vector Machines [9, 6], Naive Bayes classifier [29] etc. A number of softwares, web-servers and prediction methods are available in the literature for DNA-Binding protein prediction. Among these are: DNABinder [9], DNA-Prot [8], iDNA-Prot [7], iDNA-Prot|dis [28], DBPPred [29], iDNAPro-PseAAC [6], PseDNA-Pro [33], Kmer1+ACC [34], Local-DPP [30], etc. Kumar et al. [9] used evolutionary information from PSSM profiles with support vector machines and established a web-server called DNABinder. They compared the effectiveness of the PSSM based features with amino acid compositions, di-peptide compositions and 4-parts amino acid compositions as features.

DNA-Prot is another software proposed in [8]. They used amino-acid compositions, physio-chemical properties and secondary structure information as features and trained their model using a Random Forest classifier. Lin et al. [7] presented a web-server named iDNA-Prot where they used grey model to incorporate amino-acid sequences as features into the general form of pseudo amino acid compositions and trained their model using Random Forest classifier. Amino acid distance-pair coupling information and the amino acid reduced alphabet profile was incorporated into the general form of pseudo amino acid composition [35] by Liu et al. [28]. They also offered a freely available web-server called iDNA-Prot|dis. One of the most successful prediction methods so far is DBPPred proposed in [29]. They used a wrapper based best first feature selection technique to select the optimal set of features. They used features based on amino-acid compositions, PSSM scores, secondary structures and relative solvent accessibility and trained their model using Random Forest and Gaussian Naive Bayesian classifiers.

Liu et al. [6] presented iDNAPro-PseAAC as a web server. They used evolutionary information as features by using profile-based protein representation and selected a set of 23 features by discriminant analysis. Their model was trained using Support Vector Machine classifier. K-mer composition[36] and auto-cross covariance transformation was used in [34] in a subsequent work. Their method, trained by Support Vector machine, is known as K-mer1+ACC in the literature. They also developed another server called PseDNA-Pro [33]. PseDNA-Pro used amino acid composition, pseudo amino acid composition and physico-chemical distance transformation based features to train their model. Wei et al. proposed Local-DPP [30] by using

Random Forest classifier on local pseudo position specific scoring matrix features. Among other recent works are SVM-PSSM-DT [37], PNImodeler [38], CNNsite [39], BindUP [40] etc.

In this paper, we propose iDNAProt-ES, **i**dentification of **DNA**-binding **Prot**eins using **E**volutionary and **S**tructural Features. In our proposed method, a number of novel features have been exploited that are derived from sequence based evolutionary information and structural information of a given protein to train a support vector machine with linear kernel. We used recursive feature elimination technique to reduce the number of features and to derive an optimal set of features for DNA-binding protein prediction. We have tested our method on standard benchmark datasets. Experimental results show that iDNAProt-ES significantly outperforms other state-of-the-art methods and thus have potentials to be used as a DNA-binding protein prediction tool.

Chapter 3

Materials & Methods

In this chapter the different techniques we used in our thesis, will be presented to give a better understanding on how they were employed to perform the experiments. We will describe about the datasets, system overview, feature extraction method, feature selection method, description of the classifier and performance evaluation.

3.1 Datasets

We require a set of reliable benchmark datasets in order to develop an effective predictor using suitable classification algorithms and feature sets. Any dataset consists of positive and negative samples and can be formally denoted as follow.

$$\mathbb{S} = \mathbb{S}^+ \cup \mathbb{S}^- \tag{3.1}$$

Here $\mathbb{S}^+$ represents the set of positive instances or DNA-binding proteins and $\mathbb{S}^-$ denotes the negative samples or non DNA-binding proteins. In this paper, we have selected to use two datasets that are extensively used in the literature of DNA-binding protein prediction [33, 30, 28, 41, 6]. The first dataset, that we refer to as the *benchmark dataset* throughout this thesis was first introduced in [28]. The DNA-binding proteins were extracted from

Protein Database (PDB) [42] with the mmCIF keyword of DNA-binding protein using the advanced search interface. The query results were first filtered by removing all the sequences with length less than 50. The proteins of length under 50 often represents fragments. Then, all the protein sequences with non-standard symbol 'X' were removed.

First one is the Benchmark dataset, constructed based on the latest version of protein Data Bank (PDB) [42]. The datasets consists of 525 DNA-binding proteins and 550 non DNA-binding proteins. Protein sequences are selected based on some criterias i.e, Protein length less than 50 amino acids were removed the sequence similarity less than 25% between any two proteins were cut off by using PISCES [43] and finally proteins are randomly selected and combined.

Second one is the Independent test dataset PDB186 constructed by Lou et al [29]. Here, 93 proteins are DNA-binding proteins and 93 proteins are non-DNA-binding proteins. The NCBIs BLASTCLUST [44] was used to remove those proteins from the dataset that have more than 25% sequence identity to any protein within a same subset of the PDB186 dataset to avoid the homology bias. We have taken the datasets from Lou et al[29].

Table 3.1: Summary of Benchmark Datasets explored in this paper

Benchmark	**+ve protein**	**-ve protein**	**Total**	**Ref**
Independent dataset	93	93	186	[29]
Benchmark dataset	520	550	1070	[42]

3.2 System Overview

To establish a novel feature set and good predictor we first collected two benchmark datasets, extracted features from the data sets which are able to discriminate the DNA-binding proteins, develop the list of reduced features from the global set of features which can contribute to improve prediction accuracy, selected and developed powerful classification algorithms to per-

form prediction, performed cross-validation tests to evaluate the accuracy of the predictor.

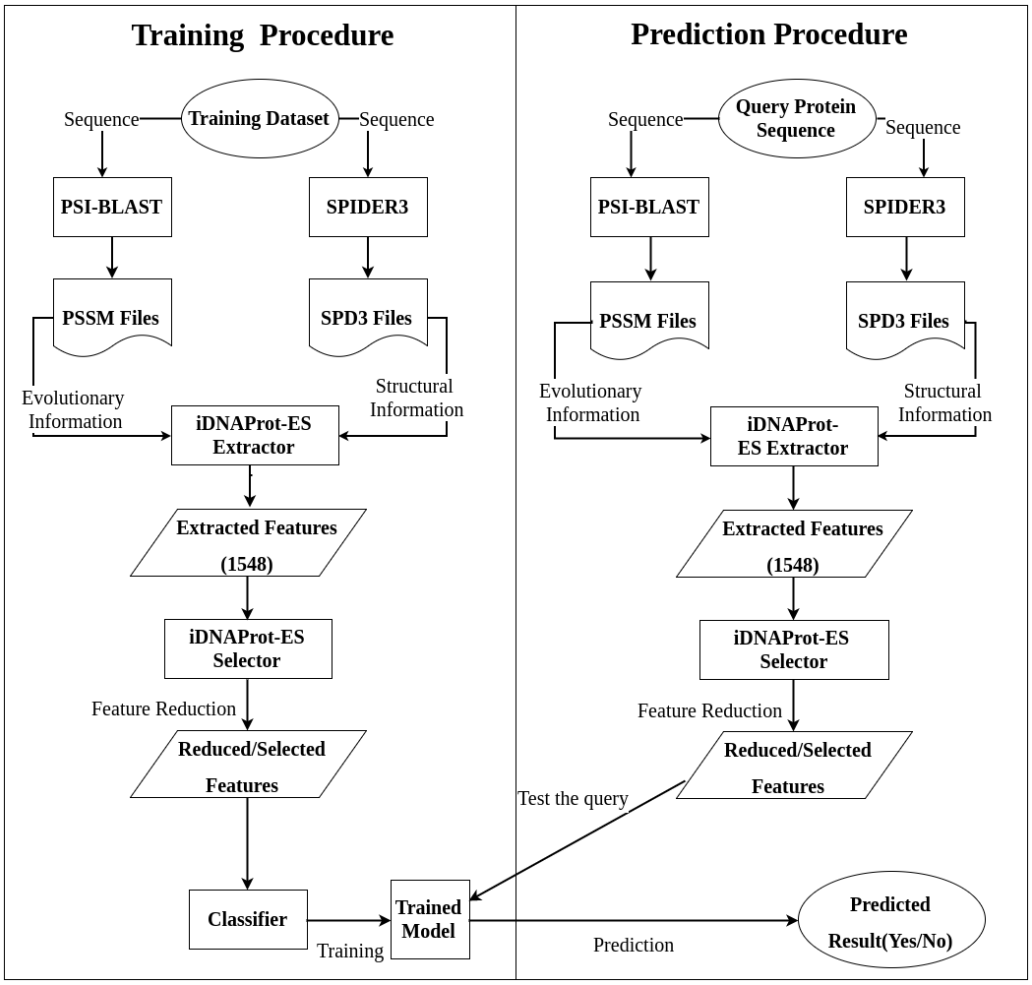

Figure 3.1: System flow diagram of iDNAProt-ES showing the training and prediction procedure as flowchart.

The framework of our proposed method iDNAProt-ES is depicted in the Figure 3.1. There are two phases in the framework for prediction: *training phase* and *prediction phase*. In training phase, at first a training dataset is selected. Next, each protein sequence from the training dataset is then passed to the PSI-BLAST [44] and SPIDER3 [45] softwares, that provide

two output files PSSM and SPD3 respectively. PSSM file is responsible for evolutionary information and SPD3 is responsible for structural information. These two files are then passed to the iDNAProt-ES feature extractor, which extract 14 sets of features. These 14 feature sets contains 1548 sub-features in total. Then sub-features (1548) are passed to the iDNAProt-ES feature selector to reduced the features to improve the prediction accuracy. Now we can get the list of reduced feature set. The reduced features are used to train a model using support vector machine classifier and stored later for prediction.

In the prediction phase, iDNAProt-ES first takes a query protein sequence and pass it to the PSI-BLAST and SPIDER3 to generate two output files PSSM and SPD3, respectively similarly to the training phase. These two files are then used by the feature extractor and feature selector of iDNAProt-ES. The reduced features are passed to the previously saved model in training phase to predict whether the protein is DNA-binding or not. These phase takes very little time compared to the training phase.

3.3 Feature Extraction

Different types of feature extraction methods are used in the literature of DNA-binding protein prediction. These include: pseudo position specific scoring matrix based features [30], pseudo amino acid composition proposed by Chou's[46] physicochemical distance transformation [33] etc. In this study, we explore evolutionary and structural information embedded in the protein sequences as features. Protein sequences are used to fetch evolutionary information extracted as PSSM (Position-Specific Scoring Matrix) files generated by PSI-BLAST [44]. In addition to that, structural information are extracted from the SPD files, output of SPIDER2 [47] software. Following sections describes the feature extraction in detail.

3.3.1 PSSM Based Features

We used evolutionary information from PSSM files generated using three iterations of the PSI-BLAST algorithm [44] using the non-redundant database (NR) provided by NCBI. The cut-off threshold value of E was set to 0.001. PSSM file returns the log-odds of the substitution probabilities of a given protein at each position for all possible amino-acid symbols after the alignment [48]. This is a $L \times 20$ matrix which we refer in this paper as *PSSM matrix*. Suppose we are given a protein sequence P consisting of L amino acid residues as follow.

$$P = R_1 R_2 R_3 .....R_L \tag{3.2}$$

The frequency profile of P generated by the PSI-BLAST [44] and matrix M can be represented as:

$$\mathbb{M} = \begin{Bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,L} \\ m_{2,1} & m_{2,2} & \cdots & m_{1,L} \\ \vdots & \vdots & \ddots & \vdots \\ m_{20,1} & m_{20,2} & \cdots & m_{20,L} \end{Bmatrix} \tag{3.3}$$

Here 20 is the number of standard amino acids; $m_{i,j}$ is the target frequency representing the log-odds probability of amino acid i (i = 1, 2, , 20) appearing in sequence position j (j = 1, 2, 3,... L) of protein P during evolutionary process. We first normalize the pssm matrix using the procedure proposed in [46] for protein sub-cellular localization. After normalization, we generated five groups of features from the normalized PSSM matrix. We will denote the normalized matrix throughout this section as N which is a two dimensional matrix of dimension $L \times 20$. The features generated from PSSM file are enumerated in the following:

1. **Amino acid composition** : The PSSM file is used to generate a consensus sequence. A consensus sequence is built by taking the amino-acid with the highest substitution probability or frequency in the PSSM

matrix at each position. Amino Acid composition then counts the occurrences of each amino-acid residue and normalizes by the length of the protein sequence.

$$AAC_j = \frac{1}{L}\sum_{i=1}^{L} aa(i,j), 1 \leq i \leq 20 \tag{3.4}$$

Here,

$$aa(i,j) = \begin{cases} 1, \text{ if } s_j = a_i \\ 0, \text{else} \end{cases}$$

where s_j is an amino acid in the protein sequence and a_i is one of the 20 different amino-acid symbols [49].

2. **Dubchak features:** Theses features were previously used for protein fold recognition [50] and protein sub-cellular localization [46]. They group the amino-acid residues according to various physico-chemical properties polarity, solvability, hydro-phobicity etc and calculates the composition, transition and distribution of these groupings. The size of the feature vector is 105.

3. **PSSM Bigram:** PSSM bigram represents the transition probabilities of two adjacent amino-acid residue positions. These features are previously used in solving protein sub-cellular localization and protein fold recognition [50, 46] and defined as below:

$$\text{PSSM-bigram}(k,l) = \frac{1}{L}\sum_{i=1}^{L-1} N_{i,k}N_{i+1,l}(1 \leq k \leq 20, 1 \leq l \leq 20) \tag{3.5}$$

4. **PSSM 1-lead Bigram[?]:** PSSM 1-lead bigram is defined as the transition probabilities of the amino-acid residue positions at 1 distance or separation. It can be formally defined as:

$$\text{PSSM-1-lead-bigram}(k,l) = \frac{1}{L}\sum_{i=1}^{L-2} N_{i,k}N_{i+2,l}(1 \leq k \leq 20, 1 \leq l \leq 20) \tag{3.6}$$

5. **PSSM Composition:** PSSM composition is created by taking the normalized sum of the values in each of the columns of the PSSM matrix [46]. Each column of the PSSM matrix represents one of the 20 amino-acid residues. It is defined as:

$$PSSM - Composition(k,l) = \frac{1}{L}\sum_{i=1}^{L-1} N_{i,j}(1 \leq j \leq 20) \tag{3.7}$$

6. **PSSM Auto-Covariance:** Auto-Covariance of PSSM is a feature [46, 51] depending of a distance factor, DF as parameter. In this study we used, DF = 10. The feature is formally defined as:

$$\text{PSSM-Auto-Covariance}(k,j) = \frac{1}{L}\sum_{i=1}^{L-k} N_{i,j}N_{i+k,j}(1 \leq j \leq 20, 1 \leq k \leq DF) \tag{3.8}$$

7. **PSSM Segmented Distribution:** Previously, the segmented distribution of the PSSM matrix proposed in [52] was used as feature for sub-cellular localization of proteins in [53]. The idea is to find the distribution of the values in the PSSM matrix column wise by calculating the partial sums columnwise starting from the first row and the last row and iterating until the partial running sum is F_p % of the total sum. The details of the procedure for this feature generation can be found in [52, 53, 54]. In this paper, we used $F_p = 5, 10, 25$.

3.3.2 SPIDER Based Features

We used SPIDER2 [47], a freely available software that provides information on accessible surface area, torsion angles, structure motifs in each amino acid residue position. We then extract a novel set of features from the information provided by SPIDER2 as SPD file. The feature extraction is enumerated here in details:

1. **Secondary Structure Occurence:** There are three types of structural motifs in proteins: α-helix (H), β-sheet (E) and random coil (C).

Secondary Structure Occurrence is the count or frequency of each type present in mino-acid residue positions.

$$\text{SS-Occurence}(i) = \sum_{j=1}^{L} sm_{ij}, 1 \leq i \leq 3 \tag{3.9}$$

Here, L is the length of the protein and

$$sm_{ij} = \begin{cases} 1, \text{ if } SS_j = \mu_i \\ 0, \text{else} \end{cases}$$

where SS_j is the structural motif at position j of the protein sequence and μ_i is one of the 3 different motif symbols.

2. **Secondary Structure Composition:** This feature is secondary structure motif occurrence normalized by the length of the phage protein length. This is similar to the amino-acid composition except that here we are taking the count of motif symbols in stead of amino-acid symbols.

$$\text{SS-Occurence}(i) = \frac{1}{L} \sum_{j=1}^{L} sm_{ij}, 1 \leq i \leq 3 \tag{3.10}$$

Here, L is the length of the protein and

$$sm_{ij} = \begin{cases} 1, \text{ if } SS_j = \mu_i \\ 0, \text{else} \end{cases}$$

where SS_j is the structural motif at position j of the protein sequence and μ_i is one of the 3 different motif symbols.

3. **Accessible Surface Area Composition:** The accessible surface area composition is the normalized sum of accessible surface area defined by:

$$\text{ASA-Composition} = \frac{1}{L} \sum_{i=1}^{L} ASA(i) \tag{3.11}$$

4. **Torsional Angles Composition:** For four different types of torsional angles: ϕ, ψ, τ and θ we first convert each of them into radians from degree angles and then take sign and cosine of the angles at each residue position. Thus we get a matrix of dimension $L \times 8$. We denote this matrix by T for torsional angles. Torsional angles composition is defined as:

$$\text{Torsional-Angles-Composition(k)} = \frac{1}{L}\sum_{i=1}^{L} T_{i,k}(1 \leq k \leq 8) \tag{3.12}$$

5. **Structural Probabilities Composition:** Structural probabilities for each position of the amino-acid residue are given in spd3 file as a matrix of dimension $L \times 3$. We denote it by P. Structural probabilities composition is defined as:

$$\text{Structural-Probabilities-Composition(k)} = \frac{1}{L}\sum_{i=1}^{L} P_{i,k}(1 \leq k \leq 3) \tag{3.13}$$

6. **Torsional Angles Bigram:** Bigram for the torsional angles is similar to that of PSSM matrix and defined as:

$$\text{Torional-angles-bigram}(k,l) = \frac{1}{L}\sum_{i=1}^{L-1} T_{i,k}T_{i+1,l}(1 \leq k \leq 8, 1 \leq l \leq 8) \tag{3.14}$$

7. **Structural Probablities Bigram:** Bigram of the structural probabilities is similar to that of PSSM matrix and defined as:

$$\text{Structural-Probabilities-bigram}(k,l) = \frac{1}{L}\sum_{i=1}^{L-1} P_{i,k}P_{i+1,l}(1 \leq k \leq 3, 1 \leq l \leq 3) \tag{3.15}$$

8. **Torsional Angles Auto-Covariance:** This feature is also derived from torsional angles and defined as:

$$\text{Torsional-Angles-Auto-Covariance}(k,j) = \frac{1}{L}\sum_{i=1}^{L-k} T_{i,j}T_{i+k,j}(1 \leq j \leq 8, 1 \leq k \leq DF) \tag{3.16}$$

9. **Structural Probablities Auto-Covariance:** This feature is also derived from structural probabilities and defined as:

$$\text{Structural-Probabilities-Auto-Covariance}(k,j) = \frac{1}{L}\sum_{i=1}^{L-k} P_{i,j}P_{i+k,j}(1 \leq j \leq 3, 1 \leq k \leq DF) \tag{3.17}$$

The features generated and used in this paper are summarized in Table 3.2.

3.4 Feature Selection

We worked and experimented on following two types of feature selection or elimination method. Both methods described in the next section.

i Subset Feature Selection

ii Recursive Feature Elimination

Feature Name	Feature Type	Feature Vector Size
Amino acid composition	Evolutionay(PSSM)	20
Dubchak feature	Evolutionay(PSSM)	105
Bigram	Evolutionay(PSSM)	400
PSSM composition	Evolutionay(PSSM)	20
PSSM auto covariance	Evolutionay(PSSM)	200
One lead bigram	Evolutionay(PSSM)	400
Segmented distribution	Evolutionay(PSSM)	200
Secondary structure composition	Structural(SPD3)	3
Secondary structure occurrence	Structural(SPD3)	3
ASA, Angle occurrence, probability of CHE	Structural(SPD3)	12
Bigram of angle sine cosine	Structural(SPD3)	64
Angles auto covariance	Structural(SPD3)	80
Bigram probabilities	Structural(SPD3)	9
Probabilities auto covariance	Structural(SPD3)	30

Table 3.2: Summary of evolutionary and structural features used in this paper.

3.4.1 Subset Feature Selection

In this study, we have extracted 1548 features. By using these large number of features our prediction accuracy didn't show satisfactory result. So we experimented with reducing features. The objectives of feature reduction is to avoid over-fitting, improve model performance and improve prediction accuracy.In the literature, genetic algorithms have been applied for feature reduction techniques like tree-based feature selection [29], Fast Correlation Based Filter [55] and clustering based feature selection [**?**] etc. In this study we used two types of feature reduction techniques[**?**]. One is GreedyStepwise[56, **?**] and another one is Best-First Search (BFS) [56]. GreedyStepwise [56] is a searching method that performs a greedy forward or backward search through the space of attribute subsets.

3.4.2 Recursive Feature Elimination

As the number of features extracted is large, we apply feature reduction to derive an optimal set of features for DNA-binding protein prediction. Previously several feature elimination techniques including correlation-based feature subset selection method [8], tree-based feature selection [29], best-first greedy feature selection [29], etc are found in the literature. In this paper, we have used Recursive feature elimination (RFE) which was first proposed in [57]. The algorithm is depicted as pseudo-code in Algorithm 1. This algorithm uses backward correlation based feature elimination technique. This algorithm starts with a dataset $\mathbb{D}$, a classifier $\mathbb{C}$ and k the number of reduced features as parameter. In each iteration of the algorithm, the dataset is used to train a model, $\mathbb{M}$ and based on that the lowest ranked feature is removed. The dataset is then transformed using the resulting features. This process is continues until the number of features is equal to k.

Recursive Feature Elimination Algorithm

Algorithm 1: RecursiveFeatureElimination($\mathbb{D}$, $\mathbb{C}$, k)

$\mathbb{D}' \leftarrow \mathbb{D}$
$\mathbb{F} = \mathbb{D}.extractAllFeatures()$
while $|\mathbb{F}| > k$ **do**
    $\mathbb{M} \leftarrow \mathbb{C}.train(\mathbb{D}')$
    $\mathbb{F}.computeRanks(\mathbb{M})$
    $f_r \leftarrow \mathbb{F}.selectLowestRank()$
    $\mathbb{F} \leftarrow \mathbb{F} - \{f_r\}$
    $\mathbb{D}' = transform(\mathbb{D}, \mathbb{F})$
end
return $\mathbb{D}'$

3.5 Description of the Classifier

To investigate the performance of our proposed approaches, several classification techniques such as, Logistic Regression, Random Forest, Naive Bayes, AdaBoost.M1, SGD and SVM are used in this study.

3.5.1 Logistic Regression:

Logistic regression tries to learn a single line from the given data to classify data which works as the decision boundary for the binary classes. In logistic regression, one requires to learn the co-efficients or weights those are parameters of the decision line. A sigmoid function is typically used as the decision rule. Gradient descent algorithms are generally used to learn the parameters.

3.5.2 Random Forest:

Random Forest is a classification technique by ensemble learning method. Random forests [58] combines bootstrap aggregating and random feature selection techniques to produce several tree predictors. The decision is a combination of the predictions by the tree classifiers.

3.5.3 Naive Bayes:

Naive Bayes [59] classifier is built on a special type of Bayesian network with only two levels of random variables. The decision variable is put in the root of the network and the feature variables are used as children dependent on it. It is built with the powerful independence assumption among features. Parameters learning is done by maximum likelihood estimator[60].

3.5.4 SGD:

SGD is a Stochastic Gradient Descent model. It is implemented for learning of different linear models i.e., SVM, logistic regression, squared loss, Huber loss and epsilon-insensitive loss linear regression. In our case we used binary class SVM.

3.5.5 BayesNet:

BayesNet or Bayes network is a probabilistic DAG model. DAG represents directed acyclic graph. BayesNet are adaptable because it allows to update all the probabilities in net for every single changes.

3.5.6 AdaBoost:

AdaBoost technique is introduced by Freund and Schapire[61] for multi class classification. It is a machine learning meta-algorithm. It may be used combined with other learning algorithms to enhance accuracy. It creates a strong classifier from the ensemble of different weak classifiers[62].

3.5.7 Support Vector Machine:

We have used Support vector machine (SVM) as the classifier in our method, iDNAProt-ES. Support Vector Machines (SVM) [63, 64] construct a separating hyper-plane to maximize the margin between the positive and negative instances. The nearest two points in the hyper-plane are called support vectors. SVM first constructs a hyper-plane based on the training dataset, and then maps an input vector from the input space into a vector in a higher dimensional space, where the mapping is determined by a kernel function. A trained SVM can output a class label (in our case, DNA-binding protein or non DNA-binding protein) based on the mapping vector of the input vector. There are a number of popular kernels. In this paper, we explore three kernel functions as described below:

1. The Linear kernel function can be defined as

$$K(X_i, X_j) = X_i X_j \tag{3.18}$$

2. The (Gaussian) radial basis function kernel, or RBF kernel can be defined as

$$K(X_i, X_j) = exp(-\gamma(||X_i - X_j||)^2) \tag{3.19}$$

3. The Sigmoid kernel function can be defined as

$$K(X_i, X_j) = \tanh(\gamma.X_i X_j) + r) \tag{3.20}$$

Here γ and r are the kernel parameters. Gamma must be greater than 0. The best kernel was the linear kernel with the parameters, $C = 1000$ and $\gamma = 0.01$.

3.6 Performance Evaluation

Many comparison metrics are used in the literature to compare the performance of different algorithms that employs supervised learning for prediction [53, 65, 66, 67]. Among them most important are: accuracy (Acc), sensitivity (Sn), specificity (Sp), Mathew's correlation co-efficient (MCC), area under Receiver Operating Characteristic curve (auROC) curve (AUC) and area under precision recall curve (auPR).

For a binary classifier, let us assume TP is the number of true positives or positive samples correctly classified by the classifier, TN is the number of true negatives or correctly classified negative samples, FP is the number of false positives or incorrectly classified negative samples as positive and FN is the number of false negatives or positive examples incorrectly classified as negatives. The sensitivity of a classifier is the true positive rate defined by the following equation: ,

$$Sen = \frac{TP}{TP + FN} \tag{3.21}$$

It denotes the predictor performance to correctly predict DNA-Binding Protein and the metrics vary from 0 to 1. Higher the value of sensitivity, the

better the predictor is to predict DNA-Binding Protein. Specificity is 1-false positive rate. It is define as in the following equation:

$$Spc = \frac{TN}{TN + FP} \tag{3.22}$$

Accuracy is the ratio of correctly classified instances to the total number of instances in a dataset defined as in following:

$$Acc = \frac{TP + TN}{TP + FN + TN + FP} \tag{3.23}$$

It also varies from 0 to 1. The higher the value is more accurate the classifier or predictor is. Mathew's correlation coefficient (MCC) is another measure of the quality of the classification model. It varies from -1 to +1 which respectively denotes negative classification correlation and positive classification correlation. MCC is defined as in the following equation:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.24}$$

AUC and auPR are two other important evaluation metrics. AUC is the area under the ROC curve which is a plot of true positive rate against false positive rate at different thresholds. auPR is the area under the precision vs recall curve. Here, precision is the ratio of true positives to the total number of predictions as positives and recall is same as the sensitivity.

Chapter 4

Result & Discussion

This chapter presents the results obtained from the experiments with the iDNAProt-ES investigation. The experiments carried out on two different processes, one is Subset Feature Selection and another one is Recursive Feature Elimination. All the experimental results on above both processes are conducted here on two different datasets. We also present a brief description on our publicly available web server implementation.

4.1 Experiments with Subset Feature Selection

Feature Selection or attribute selection [**?**] is a process by which we can automatically search for the best subset of attributes in our dataset. Attribute Evaluator is a method by which attribute subsets are assessed. The Search Method is the structured way in which the search space of possible attribute subsets is navigated based on the subset evaluation.

Our method is implemented by using two types search method "GreedyStepWise" and "BestFirst"; and one types of Attribute Evaluator "CfsSubsetEval" from Weka. GreedyStepWise[**?**] Uses a forward (additive) or backward (subtractive) step-wise strategy to navigate attribute subsets. BestFirst [**?**]

Uses a best-first search strategy to navigate attribute subsets. CfsSubsetEval [**?**] Values subsets that correlate highly with the class value and low correlation with each other. Our total features 1548 reduced to 36 set feature by using "GreedyStepWise" and "CfsSubsetEval". The total features also reduced to 53 by using "BestFirst" and "CfsSubsetEval". All the experiments have been conducted by using Waikato Environment for Knowledge Analysis (WEKA) [56]. The reduced set of 36 features and reduced set of 53 features enlisted in the Table A.1 and Table B.1 respectively.

Comparison among different classifier

To compare the performance of different classifier we used benchmark dataset and performed 10 fold cross validation test and reported accuracy, sensitivity, specificity and MCC in Table 4.1 and Table 4.2 by using search method GreedyStepWise and BestFirst respectively. From the result we can observe that SMO with RBF kernel achieves the best accuracy for benchmark dataset with reduced set of 36 features (Table 4.1). But Logistic classifier [68] shows the best accuracy for the benchmark dataset with a reduced set of 53 features (Table 4.2). From Tables 4.1 and 4.2 we can see that the reduced set of 36 features by GreedyStepWise technique perform better than the reduced set of 53 features by BestFirst Search.

From the results shown in the tables, we notice that there were not a single classification algorithm that performed superior compared to others on both of the datasets. In the case of benchmark dataset, we found logistic regression and SVM with RBF kernel to perform the best among all the classifiers in terms of accuracy. However, when the reduced set of 36 features were used with SVM slightly better performance is achieved. Logistic regression, despite its simplicity, performed well in terms of other evaluation metrics.

We also show Receiver Operating Characteristic (ROC) curves for each of these reduced feature sets using the logistic regression classifier in Figure 4.1 and Figure 4.2. The ROC plots for other classifiers are also similar.

Classifier	Acc	Sn	Sp	MCC	auROC
Logistic	76.54%	79.30	73.70	0.530	**0.839**
BayesNet	72.05%	77.30	66.50	0.441	0.802
NaiveBayes	70.37%	**82.70**	57.30	0.415	0.781
SGD	77.19%	77.30	77.10	0.544	0.772
AdaBoost	75.51%	75.10	76.00	0.510	0.811
Random Forest	75.88%	78.00	73.70	0.517	0.831
SMO with RBF	**77.76%**	77.80	**77.70**	**0.555**	0.778

Table 4.1: Comparison of performance of different Classifiers on the benchmark dataset using 10-fold cross validation. Feature reduction using GreedyStepwise-cfsSubsetEval and reduced to 36 features.

Classifier	Acc	Sn	Sp	MCC	auROC
Logistic	**77.10%**	**80.20**	73.80	**0.542**	**0.834**
BayesNet	71.77%	77.10	66.20	0.436	0.800
NaiveBayes	71.77%	77.10	66.20	0.436	0.800
SGD	75.61%	76.00	75.20	0.512	0.756
AdaBoost	70.93%	70.90	71.00	0.419	0.785
Random Forest	75.23%	77.10	73.30	0.504	0.824
SMO with RBF	76.63%	75.10	**78.30**	0.533	0.767

Table 4.2: Comparison of performance of different Classifiers on the benchmark dataset using 10-fold cross validation. Feature reduction using BFS-CfsSubsetEval and reduced to 53 features.

Comparison with other Methods

Table 4.3 shows the comparison of our method with the other state-of-art methods applied on the independent dataset and reports the accuracy, sensitivity, specificity and MCC. We compare our predictor performance with other state-of-art in the literature: iDNAPro-PseAAC [6], iDNA-Prot [7], DNA-Prot [8], DNAbinder [9], DNABIND [10], DNA-Threader [11]. We also compare our prediction performance with other state-of art prediction performance for the benchmark dataset shows in Table 4.4. For both datasets our method showed the best prediction accuracy of other Methods.

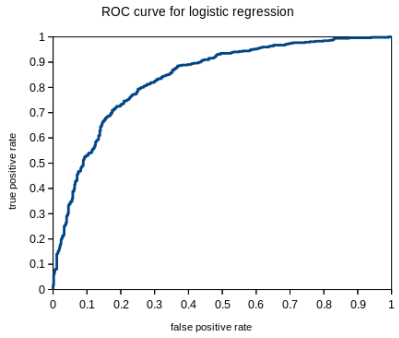

Figure 4.1: Classifier Analysis. Receiver Operation Characteristics (ROC) curve of benchmark dataset reduced set of feature-36 dataset

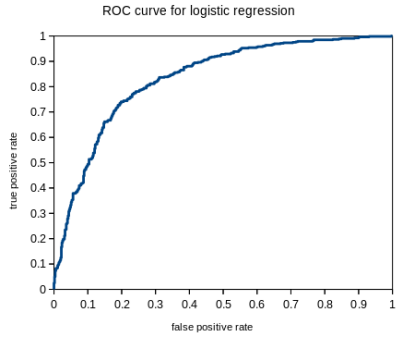

Figure 4.2: Classifier Analysis. Receiver Operation Characteristics (ROC) curve of benchmark dataset reduced set of feature-53 dataset

4.2 Experiments with RFE

In this section, we present the results of the experiments which we found in Recursive Feature Elimination (RFE). All the methods were implemented in Python language using Python3.4 version and Scikit-learn library [69] of Python was used for the implementation of machine learning algorithms. All experiments were conducted on a Computing Machine provided by CITS, United International University. The machine was equipped with 8 core processors each core having a Dell R 730 Intel Xeon Processor (E5-2630 V3)

Method	Acc	Sn	Sp	MCC	auROC
Our Method	**75.81%**	**80.60**	71.00	**0.519**	**0.822**
iDNAPro-PseAAC	69.89%	77.41	62.37	0.402	0.775
iDNA-Prot	67.20%	67.70	66.70	0.344	NA
DNA-Prot	61.80%	69.90	53.80	0.240	NA
DNAbinder	60.80%	57.00	64.50	0.216	0.607
DNABIND	67.70%	66.70	68.80	0.355	0.694
DNA-Threader	59.70%	23.70	**95.70**	0.279	NA

Table 4.3: Comparison of the results obtained by our method and the other methods on the independent dataset. The results of iDNAPro-PseAAC [6], iDNA-Prot [7], DNA-Prot [8], DNAbinder [9], DNABIND [10], DNA-Threader [11],and DBPPred[11] were obtained from[12]. 'NA' represents the unreported values.

Method	Acc	Sn	Sp	MCC	auROC
Our method	**77.76%**	77.8	77.7	0.55	0.778
iDNAPro-PseAAC	76.56%	75.62	77.45	0.53	**0.839**
DNAbinder(dimension 21)	73.95%	68.57	79.09	0.48	0.814
DNAbinder(dimension 400)	73.58%	66.47	80.36	0.47	0.815
DNA-Protc	72.55%	82.67	59.76	0.44	0.789
iDNA-Protd	75.40%	83.81	64.73	0.50	0.761

Table 4.4: Comparison of the results obtained by our method and the other methods on the benchmark dataset.

with 2.4 GHz speed and 18.5 GB memory. Each of the experiments were carried 5 times and only the average is reported as results. Table C.1 contain the list of features selected by the recursive feature elimination technique from all the combined evolutionary and structural features. The mentioned table contains the feature number and the feature group from where it comes.

Method	Acc	Sn	Sp	MCC	auROC
iDNAPro-PseAAC	76.76%	0.7562	0.7745	0.53	0.8392
DNAbinder (dimension 21)	73.95%	0.6857	0.7909	0.48	0.8140
DNAbinder (dimension 400)	73.58%	0.6647	0.8036	0.47	0.8150
DNA-Prot	72.55%	0.8267	0.5976	0.44	0.7890
iDNA-Prot	75.40%	0.8381	0.6473	0.50	0.7610
iDNA-Prot|dis	77.30%	0.7940	0.7527	0.54	0.8310
PseDNA-Pro	76.55%	0.7961	0.7363	0.53	–
Kmer1+ACC	75.23%	0.7676	0.7376	0.50	0.8280
Local-DPP	79.20%	0.8400	0.7450	0.59	–
iDNAProt-ES	**90.18%**	**0.9038**	**0.9000**	**0.8036**	**0.9412**

Table 4.5: Comparison of performance of the proposed method with other state-of-the-art predictors using jack knife test on the benchmark dataset.

Method	Acc	Sn	Sp	MCC	auROC
iDNAPro-PseAAC	69.89%	0.7741	0.6237	0.402	0.7754
iDNA-Prot	67.20%	0.6770	0.6670	0.344	–
DNA-Prot	61.80%	0.6990	0.5380	0.240	–
DNAbinder	60.80%	0.5700	0.6450	0.216	0.6070
DNABIND	67.70%	0.6670	0.6880	0.355	0.6940
DNA-Threader	59.70%	0.2370	**0.9570**	0.279	–
DBPPred	76.90%	0.7960	0.7420	0.538	0.7910
iDNA-Prot|dis	72.00%	0.7950	0.6450	0.445	0.7860
Kmer1+ACC	70.96%	0.8279	0.5913	0.431	0.7520
Local-DPP	79.00%	**0.9250**	0.6560	**0.625**	–
iDNAProt-ES	**80.64%**	0.8131	0.8000	0.6130	**0.8434**

Table 4.6: Comparison of performance of the proposed method with other state-of-the-art predictors on the independent dataset.

4.2.1 Comparison With Other Methods

To compare the performance of our predictor iDNAProt-ES with other state-of-the-art algorithms, we first used the benchmark dataset. Using this dataset, we performed jack knife test and report accuracy, sensitivity, specificity,

MCC and auROC values in Table 4.5. We compare the results achieved by iDNAProt-ES with previous state-of-the-art methods in the literature: DNABinder [9], DNA-Prot [8], iDNA-Prot [7], iDNA-Prot|dis [28], DBP-Pred [29], iDNAPro-PseAAC [6], PseDNA-Pro [33], Kmer1+ACC [34] and Local-DPP [30]. The results reported in this paper for these methods are taken from [30, 6].

The best values in Table 4.5 are shown in bold faced font. For the benchmark dataset our method iDNAProt-ES significantly outperforms the previous state-of-th-art in terms of all the evaluation metrics used. Accuracy of iDNAProt-ES is 90.18% compared to the previous best 79.20% by Local-DPP [30]. The higher MCC value and auROC also depicts the effective ness of our method.

To assess the performance further, we applied iDNAProt-ES on the independent dataset introduced in [29]. Here, we used the same model trained using iDNAProt-ES on the benchmark dataset and tested using the independent dataset. We report the performance metrics in Table 4.6 for the independent dataset. Here too, the best values are shown in bold faced font. We could notice that our algorithm is showing slightly better performance in terms of accuracy and auROC compared to the other state-of-the-art algorithms. However, the sensitivity, specificity and MCC values are not the best, but comparable to the other methods.

Figure 4.3: Effect of number of features selected on the accuracy on the benchmark dataset

Method	Acc	Sn	Sp	MCC	auROC	auPR
RFE	**88.87%**	**0.8945**	**0.8826**	**0.7788**	**0.9391**	**0.8828**
Tree Based Method	70.93%	0.7627	0.6480	0.4196	0.7775	0.6470
Sparse Elimination	75.98%	0.7727	0.7461	0.5210	0.8308	0.7464
No Feature Selection	74.01%	0.7581	0.7211	0.4835	0.8224	0.7242

Table 4.7: Comparison of performance of different feature selection methods on the benchmark dataset using 10-fold cross validation.

4.2.2 Effect of Feature Selection

In this section, we show the effect of the feature selection algorithm that we used. For this experiment we used 10-fold cross validation on both of the datasets to find the optimal set of features using recursive feature elimination technique. We varied the number of features from $25 \cdots 100$ using the recursive feature elimination technique for two SVM kernels: sigmoid and linear. The highest accuracy was found when the number of reduced features were set to 86. Figure 4.3 shows the plot of accuracy against the number of reduced features using recursive feature selection algorithm using two classifiers.

Color map of the rankings of the features as ranked by the RFE algorithm is given in 4.4. This color map depicts the distribution of selected features over all the features. Selected features include Dubchuck features, PSSM bigram, PSSM Auto-Covariance, PSSM 1-lead bigram and PSSM segmented distribution from the evolutionary group of features extracted for PSSM and the rest of the features were structural features generated by SPIDER2. It reveals the importance of both type of features: evolutionary and structural.

We then compared the performance of this feature selection technique with other feature selection techniques: tree based method [70] and randomized sparse elimination [71, 72] and no feature elimination. We performed 10-fold cross validation for these experiments too and applied different feature elimination techniques on the benchmark dataset and report the results in Table 4.7.

Here too, we show the best values achieved in bold faced fonts. We could

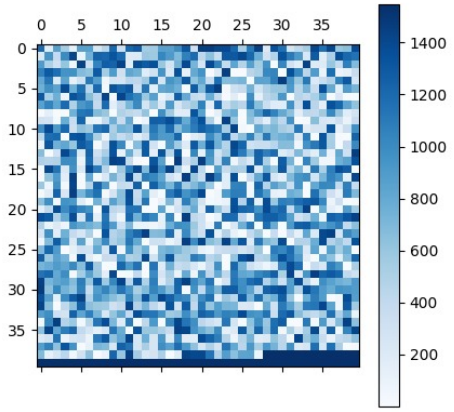

Figure 4.4: Color map showing the importance or ranking of the features on the benchmark dataset

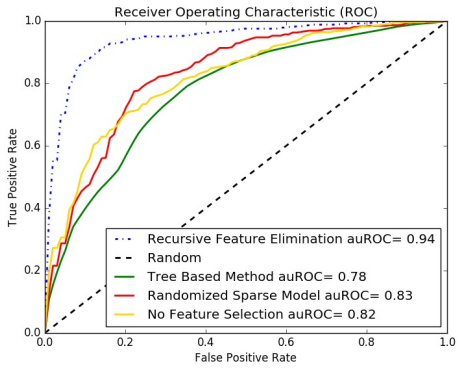

Figure 4.5: Receiver Operating Characteristic (ROC) curve of different feature selection methods on the benchmark dataset.

easily note that recursive feature elimination technique was the best among the feature elimination techniques that were used in the experiments. We also show the Receiver Operating Curve (ROC) for each of these methods for the benchmark dataset in Figure 4.5.

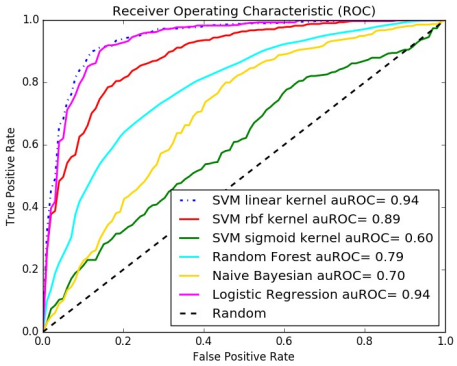

Figure 4.6: Receiver Operating Characteristic (ROC) curve of different classifiers for the benchmark dataset.

Classifier	Acc	Sn	Sp	MCC	auROC	auPR
SVM (linear kernel)	**88.87%**	**0.8945**	**0.8826**	**0.7788**	**0.9391**	**0.8828**
SVM (rbf kernel)	81.96%	0.8309	0.8076	0.6415	0.8866	0.8117
SVM (sigmoid kernel)	56.07%	0.5672	0.5538	0.1218	0.6010	0.5527
Random Forest	70.56%	0.7636	0.6442	0.4107	0.7881	0.6451
Naive Bayes	61.58%	0.7545	0.4692	0.2362	0.7005	0.4726
Logistic Regression	86.72%	0.8800	0.8538	0.7359	0.9359	0.8567

Table 4.8: Comparison of performance of different Classifiers on the benchmark dataset using 10-fold cross validation.

4.2.3 Effect of Classifier Selection

To justify the classifier selection for our algorithm, we ran another set of experiments on the benchmark dataset using 10-fold cross validation. Several classifiers were tested in the experiments: Support Vector Machine with linear kernel, Support Vector Machine with RBF kernel, Support Vector Machine with sigmoid kernel, Random Forest Classifier, Naive Bayes Classifier and Logistic Regression Classifier. The results achieved in these experiments are shown in Table 4.8.

The best values in Table 4.8 are shown in bold faced fonts. We could see

the SVM classifier with linear kernel outperformed all other classifiers. The closest competitor to linear kernel was the logistics regression classifier and the SVM with rbf kernel. We also show the ROC curve for this experiment in Figure 4.6.

4.3 iDNAProt-ES Web Server Implementation

To make the predictor iDNAProt-ES freely available for use and test we implemented a web server. This web application is freely available for use from the web address http://brl.uiu.ac.bd/iDNAProt-ES/. The predictor is very easy to use and the model is trained using the benchmark dataset. To use this site for identification of DNA-binding proteins, one has to provide two input files: PSSM file generated by PSI-BLAST [44] and a SPD file generated by SPIDER2 [47]. After these files are uploaded iDNAProt-ES, will extract features and follow a similar procedure as shown in figure 3.1. A detail guideline is provided in the website to use the predictor. Some screen shots of the web server is given in Figure 4.7 from Figure 4.11.

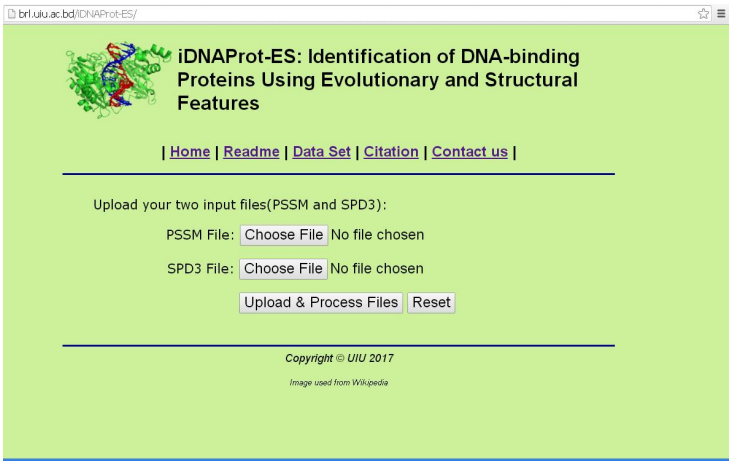

Figure 4.7: Screen shot of Web-Server homepage.

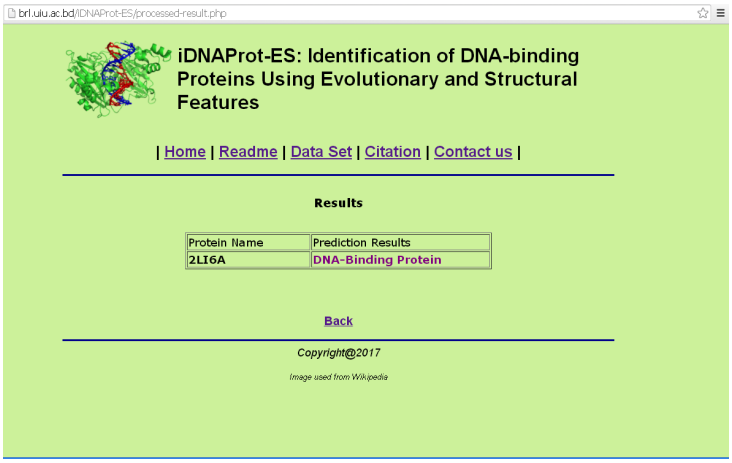

Figure 4.8: Screen shot of Web-Server Result of positive data.

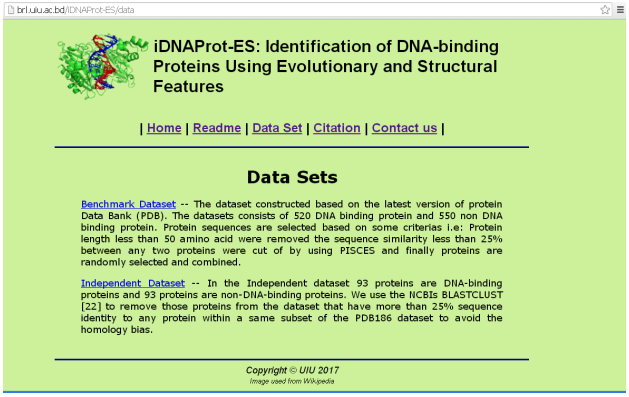

Figure 4.9: Screen shot of Web-Server Datasets.

Data and Material Availability

All the data and materials used in this paper are available at: http://brl.uiu.ac.bd/iDNAProt-ES/.

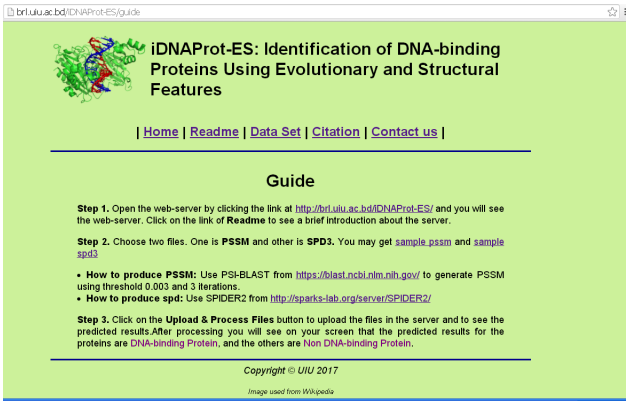

Figure 4.10: Screen shot of Web-Server Guide or Readme.

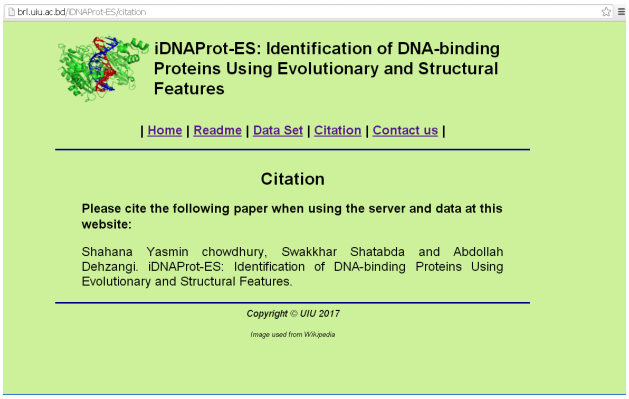

Figure 4.11: Screen shot of Journal citation.

Chapter 5

Conclusion & Future Research

5.1 Summary

In this paper, we present iDNAProt-ES, a novel prediction method for identification of DNA-binding proteins. We have used evolutionary and structural features for the classification extracted from PSSM files and SPD files generated by PSI-BLAST and SPIDER2 respectively. We also used recursive feature elimination to select an optimal set of features. The final model for prediction was developed using Support Vector Machine (SVM) with linear kernel. iDNAProt-ES was tested on a standard benchmark dataset and an independent dataset and achieved significantly improved results on both of the datasets. The method is freely available for use at: http://brl.uiu.ac.bd/iDNAProt-ES/.

The superiority of iDNAProt-ES was clearly noticeable in the experiments done in this study. The insights gained from this study are not limited to the iDNAProt-ES but could be applied to other DNA-protein interaction researches.

5.2 Limitations

Generally non-DNA-binding proteins dominate in a proteomes. So there is a chance to preferred to have high specificity to avoid over-predicting DNA-binding proteins for real-world applications. Though we have address this issue but it's not solve in our study.

In our experimental result, for the independent dataset on different our method iDNAProt-ES shows the best prediction performance 80.64%. Though the AUC gives the higher result, but other metrics such as sensitivity(Sn), specificity(Sp) and MCC is not the best in any of them in the same test.

5.3 Future Work

In future, we wish to update the prediction method by incorporating an enhanced dataset. In practice, we know that the number of negative samples are much higher than that of the positive samples. Therefore, an enhanced dataset with balancing methods could further enhance the performance of the predictor.

Appendix A

Appendix A

SL No.	Feature Number	Feature Group
1	attribute 8	Amino Acid Composition
2	attribute 23	Dubchak
3	attribute 29	Dubchak
4	attribute 33	Dubchak
5	attribute 40	Dubchak
6	attribute 42	Dubchak
7	attribute 44	Dubchak
8	attribute 46	Dubchak
9	attribute 51	Dubchak
10	attribute 52	Dubchak
11	attribute 67	Dubchak
12	attribute 68	Dubchak
13	attribute 71	Dubchak
14	attribute 86	Dubchak
15	attribute 87	Dubchak
16	attribute 92	Dubchak
17	attribute 95	Dubchak
18	attribute 97	Dubchak
19	attribute 107	Dubchak
20	attribute 114	Dubchak
21	attribute 115	Dubchak

22	attribute 116	Dubchak
23	attribute 127	Dubchak
24	attribute 539	PSSM Composition
25	attribute 773	PSSM 1 Lead Bigram
26	attribute 1088	PSSM 1 Lead Bigram
27	attribute 1224	PSSM Segmented Distrubution
28	attribute 1354	Angle Bigram
29	attribute 1355	Angle Bigram
30	attribute 1359	Angle Bigram
31	attribute 1449	ASA Occurance
32	attribute 1455	ASA Occurance
33	attribute 1511	Probabilities Auto Convariance
34	attribute 1513	Probabilities Auto Convariance
35	attribute 1515	Probabilities Auto Convariance
36	attribute 1538	Probabilities Auto Convariance

Table A.1: Reduced set of 36 features: Produced by using GreedyStepWise Search

Appendix B

Appendix B

SL No	Feature Number	Feature Group
1	attribute 8	Amino Acid Composition
2	attribute 13	Amino Acid Composition
3	attribute 23	Dubchak
4	attribute 24	Dubchak
5	attribute 29	Dubchak
6	attribute 33	Dubchak
7	attribute 35	Dubchak
8	attribute 36	Dubchak
9	attribute 40	Dubchak
10	attribute 41	Dubchak
11	attribute 42	Dubchak
12	attribute 44	Dubchak
13	attribute 51	Dubchak
14	attribute 52	Dubchak
15	attribute 53	Dubchak
16	attribute 55	Dubchak
17	attribute 67	Dubchak
18	attribute 68	Dubchak
19	attribute 71	Dubchak
20	attribute 85	Dubchak
21	attribute 86	Dubchak

22	attribute 87	Dubchak
23	attribute 88	Dubchak
24	attribute 89	Dubchak
25	attribute 92	Dubchak
26	attribute 95	Dubchak
27	attribute 97	Dubchak
28	attribute 107	Dubchak
29	attribute 114	Dubchak
30	attribute 115	Dubchak
31	attribute 116	Dubchak
32	attribute 122	Dubchak
33	attribute 127	Dubchak
34	attribute 353	PSSM Bigram
35	attribute 539	PSSM Composition
36	attribute 773	PSSM 1 Lead Bigram
37	attribute 1088	PSSM 1 Lead Bigram
38	attribute 1224	PSSM Segmented Distrubution
39	attribute 1354	Angle Bigram
40	attribute 1355	Angle Bigram
41	attribute 1359	Angle Bigram
42	attribute 1387	Angle Bigram
43	attribute 1407	Angle Bigram
44	attribute 1440	Angle Auto Covariance
45	attribute 1448	Angle Auto Covariance
46	attribute 1449	Angle Auto Covariance
47	attribute 1453	Angle Auto Covariance
48	attribute 1455	Angle Auto Covariance
49	attribute 1457	Angle Auto Covariance
50	attribute 1511	Probabilities Auto Convariance
51	attribute 1513	Probabilities Auto Convariance
52	attribute 1515	Probabilities Auto Convariance
53	attribute 1538	Probabilities Auto Convariance

Table B.1: Reduced set of 53 features: Produced by using BestFirst Search

Appendix C

Appendix C

SL No.	Feature Number	Feature Group
1	Feature_110	Dubchuck
2	Feature_142	PSSM Bigram
3	Feature_145	PSSM Bigram
4	Feature_146	PSSM Bigram
5	Feature_187	PSSM Bigram
6	Feature_219	PSSM Bigram
7	Feature_250	PSSM Bigram
8	Feature_264	PSSM Bigram
9	Feature_266	PSSM Bigram
10	Feature_272	PSSM Bigram
11	Feature_277	PSSM Bigram
12	Feature_279	PSSM Bigram
13	Feature_380	PSSM Bigram
14	Feature_321	PSSM Bigram
15	Feature_323	PSSM Bigram
16	Feature_324	PSSM Bigram
17	Feature_330	PSSM Bigram
18	Feature_335	PSSM Bigram
19	Feature_342	PSSM Bigram
20	Feature_346	PSSM Bigram
21	Feature_373	PSSM Bigram

22	Feature_386	PSSM Bigram
23	Feature_393	PSSM Bigram
24	Feature_427	PSSM Bigram
25	Feature_444	PSSM Bigram
26	Feature_459	PSSM Bigram
27	Feature_475	PSSM Bigram
28	Feature_479	PSSM Bigram
29	Feature_496	PSSM Bigram
30	Feature_522	PSSM Bigram
31	Feature_574	PSSM Auto-Covariance
32	Feature_577	PSSM Auto-Covariance
33	Feature_590	PSSM Auto-Covariance
34	Feature_593	PSSM Auto-Covariance
35	Feature_601	PSSM Auto-Covariance
36	Feature_610	PSSM Auto-Covariance
37	Feature_622	PSSM Auto-Covariance
38	Feature_640	PSSM Auto-Covariance
39	Feature_641	PSSM Auto-Covariance
40	Feature_666	PSSM Auto-Covariance
41	Feature_690	PSSM Auto-Covariance
42	Feature_694	PSSM Auto-Covariance
43	Feature_703	PSSM Auto-Covariance
44	Feature_714	PSSM Auto-Covariance
45	Feature_736	PSSM Auto-Covariance
46	Feature_742	PSSM Auto-Covariance
47	Feature_754	PSSM 1-lead Bigram
48	Feature_767	PSSM 1-lead Bigram
49	Feature_775	PSSM 1-lead Bigram
50	Feature_799	PSSM 1-lead Bigram
51	Feature_808	PSSM 1-lead Bigram
52	Feature_815	PSSM 1-lead Bigram
53	Feature_858	PSSM 1-lead Bigram
54	Feature_862	PSSM 1-lead Bigram
55	Feature_908	PSSM 1-lead Bigram
56	Feature_911	PSSM 1-lead Bigram
57	Feature_928	PSSM 1-lead Bigram

58	Feature_946	PSSM 1-lead Bigram
59	Feature_951	PSSM 1-lead Bigram
60	Feature_974	PSSM 1-lead Bigram
61	Feature_992	PSSM 1-lead Bigram
62	Feature_1004	PSSM 1-lead Bigram
63	Feature_1029	PSSM 1-lead Bigram
64	Feature_1040	PSSM 1-lead Bigram
65	Feature_1064	PSSM Segmented Distribution
66	Feature_1075	PSSM Segmented Distribution
67	Feature_1089	PSSM Segmented Distribution
68	Feature_1090	PSSM Segmented Distribution
69	Feature_1094	PSSM Segmented Distribution
70	Feature_1104	PSSM Segmented Distribution
71	Feature_1119	PSSM Segmented Distribution
72	Feature_1131	PSSM Segmented Distribution
73	Feature_1134	PSSM Segmented Distribution
74	Feature_1219	PSSM Segmented Distribution
75	Feature_1273	PSSM Segmented Distribution
76	Feature_1283	PSSM Segmented Distribution
77	Feature_1308	PSSM Segmented Distribution
78	Feature_1373	Angles Bigram
79	Feature_1376	Angles Bigram
80	Feature_1386	Angles Bigram
81	Feature_1392	Angles Bigram
82	Feature_1393	Angles Bigram
83	Feature_1486	Angles Auto-Covariance
84	Feature_1519	Probabilities Auto-Covariance
85	Feature_1528	Probabilities Auto-Covariance
86	Feature_1537	Probabilities Auto-Covariance

Table C.1: The list of features selected by the Recursive Feature Elimination (RFE) technique.

Bibliography

[1] Dna, https://en.wikipedia.org/wiki/dna (aug 2017).

[2] Protein structure, http://www.particlesciences.com/news/technical-briefs/2009/protein-structure.html.

[3] Central dogma of molecular biology, https://en.wikipedia.org/wiki/central_dogma_of_molecular_biology/ (aug 2017).

[4] Protein_biosynthesis (biology), https://en.wikipedia.org/wiki/protein_biosynthesis (aug 2017).

[5] Dna binding protein https://en.wikipedia.org/wiki/dna-binding_protein.html (aug 2017).

[6] Bin Liu, Shanyi Wang, and Xiaolong Wang. Dna binding protein identification by combining pseudo amino acid composition and profile-based protein representation. *Scientific reports*, 5:15479, 2015.

[7] Wei-Zhong Lin, Jian-An Fang, Xuan Xiao, and Kuo-Chen Chou. idna-prot: identification of dna binding proteins using random forest with grey model. *PloS one*, 6(9):e24756, 2011.

[8] K Krishna Kumar, Ganesan Pugalenthi, and PN Suganthan. Dna-prot: identification of dna binding proteins from protein sequence information using random forest. *Journal of Biomolecular Structure and Dynamics*, 26(6):679–686, 2009.

[9] Manish Kumar, Michael M Gromiha, and Gajendra PS Raghava. Identification of dna-binding proteins using support vector machines and evolutionary profiles. *BMC bioinformatics*, 8(1):463, 2007.

[10] András Szilágyi and Jeffrey Skolnick. Efficient prediction of nucleic acid binding function from low-resolution protein structures. *Journal of molecular biology*, 358(3):922–933, 2006.

[11] Mu Gao and Jeffrey Skolnick. A threading-based method for the prediction of dna-binding proteins with application to the human genome. *PLoS Comput Biol*, 5(11):e1000567, 2009.

[12] Hugh P Shanahan, Mario A Garcia, Susan Jones, and Janet M Thornton. Identifying dna-binding proteins using structural motifs and the electrostatic potential. *Nucleic Acids Research*, 32(16):4732–4741, 2004.

[13] David MJ Lilley, BD Hames, and David M Glover. *DNA-protein: structural interactions*. IRL Press at Oxford University Press Oxford:, 1995.

[14] Christoph Zimmer and Ulla Wähnert. Nonintercalating dna-binding ligands: specificity of the interaction and their use as tools in biophysical, biochemical and biological investigations of the genetic material. *Progress in biophysics and molecular biology*, 47(1):31–112, 1986.

[15] Reham Helwa and Jörg D Hoheisel. Analysis of dna–protein interactions: from nitrocellulose filter binding assays to microarray studies. *Analytical and bioanalytical chemistry*, 398(6):2551–2561, 2010.

[16] Katie Freeman, Marc Gwadz, and David Shore. Molecular and genetic analysis of the toxic effect of rap1 overexpression in yeast. *Genetics*, 141(4):1253–1262, 1995.

[17] Rahul Jaiswal, Samarendra K Singh, Deepak Bastia, and Carlos R Escalante. Crystallization and preliminary x-ray characterization of the eukaryotic replication terminator reb1–ter dna complex. *Acta Crystallographica Section F: Structural Biology Communications*, 71(4):414–418, 2015.

[18] Michael J Buck and Jason D Lieb. Chip-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics*, 83(3):349–360, 2004.

[19] Gary Felsenfeld, Cecelia Trainor, and Ettore Appella. Nmr structure of a specific dna complex of zn-containing dna binding domain of gata-1. *Geophys. Res. Lett*, 20:97, 1993.

[20] Robert E Langlois and Hui Lu. Boosting the prediction and understanding of dna-binding domains from sequence. *Nucleic acids research*, page gkq061, 2010.

[21] UniProt Consortium et al. Uniprot: the universal protein knowledgebase. *Nucleic acids research*, 45(D1):D158–D169, 2017.

[22] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. Molecular biology of the cell. new york: Garland science; 2002. *Classic textbook now in its 5th Edition*, 2002.

[23] Protein, https://en.wikipedia.org/wiki/protein.html.

[24] https://www.nature.com/subjects/dna-binding-proteins.

[25] https://en.wikipedia.org/wiki/smith

[26] https://www.ncbi.nlm.nih.gov/class/structure/pssm/pssm_viewer.cgi.

[27] Huiying Zhao, Yuedong Yang, and Yaoqi Zhou. Structure-based prediction of dna-binding proteins by structural alignment and a volume-fraction corrected dfire-based energy function. *Bioinformatics*, 26(15):1857–1863, 2010.

[28] Bin Liu, Jinghao Xu, Xun Lan, Ruifeng Xu, Jiyun Zhou, Xiaolong Wang, and Kuo-Chen Chou. idna-prot— dis: identifying dna-binding proteins by incorporating amino acid distance-pairs and reduced alphabet profile into the general pseudo amino acid composition. *PloS one*, 9(9):e106691, 2014.

[29] Wangchao Lou, Xiaoqing Wang, Fan Chen, Yixiao Chen, Bo Jiang, and Hua Zhang. Sequence based prediction of dna-binding proteins based on hybrid feature selection using random forest and gaussian naive bayes. *PLoS One*, 9(1):e86703, 2014.

[30] Leyi Wei, Jijun Tang, and Quan Zou. Local-dpp: An improved dna-binding protein prediction method by exploring local evolutionary information. *Information Sciences*, 384:135–144, 2017.

[31] Guy Nimrod, Maya Schushan, András Szilágyi, Christina Leslie, and Nir Ben-Tal. idbps: a web server for the identification of dna binding proteins. *Bioinformatics*, 26(5):692–693, 2010.

[32] Shandar Ahmad, M Michael Gromiha, and Akinori Sarai. Analysis and prediction of dna-binding proteins and their binding residues based on composition, sequence and structural information. *Bioinformatics*, 20(4):477–486, 2004.

[33] Bin Liu, Jinghao Xu, Shixi Fan, Ruifeng Xu, Jiyun Zhou, and Xiaolong Wang. Psedna-pro: Dna-binding protein identification by combining chous pseaac and physicochemical distance transformation. *Molecular Informatics*, 34(1):8–17, 2015.

[34] Qiwen Dong, Shanyi Wang, Kai Wang, Xuan Liu, and Bin Liu. Identification of dna-binding proteins by auto-cross covariance transformation. In *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on*, pages 470–475. IEEE, 2015.

[35] Kuo-Chen Chou. Some remarks on protein attribute prediction and pseudo amino acid composition. *Journal of theoretical biology*, 273(1):236–247, 2011.

[36] https://en.wikipedia.org/wiki/k-mer.

[37] Ruifeng Xu, Jiyun Zhou, Hongpeng Wang, Yulan He, Xiaolong Wang, and Bin Liu. Identifying dna-binding proteins by combining support vector machine and pssm distance transformation. *BMC systems biology*, 9(1):S10, 2015.

[38] Jinyong Im, Narankhuu Tuvshinjargal, Byungkyu Park, Wook Lee, De-Shuang Huang, and Kyungsook Han. Pnimodeler: web server for inferring protein-binding nucleotides from sequence data. *BMC genomics*, 16(3):S6, 2015.

[39] Jiyun Zhou, Qin Lu, Ruifeng Xu, Lin Gui, and Hongpeng Wang. Cnnsite: Prediction of dna-binding residues in proteins using convolutional neural network with sequence features. In *Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on*, pages 78–85. IEEE, 2016.

[40] Inbal Paz, Efrat Kligun, Barak Bengad, and Yael Mandel-Gutfreund. Bindup: a web server for non-homology-based prediction of dna and rna binding proteins. *Nucleic acids research*, 44(W1):W568–W574, 2016.

[41] Bin Liu, Longyun Fang, Fule Liu, Xiaolong Wang, Junjie Chen, and Kuo-Chen Chou. Identification of real microrna precursors with a pseudo structure status composition approach. *PloS one*, 10(3):e0121501, 2015.

[42] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank, 1999–. In *International Tables for Crystallography Volume F: Crystallography of biological macromolecules*, pages 675–684. Springer, 2006.

[43] Guoli Wang and Roland L Dunbrack. Pisces: recent improvements to a pdb sequence culling server. *Nucleic acids research*, 33(suppl 2):W94–W98, 2005.

[44] Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402, 1997.

[45] Rhys Heffernan, Kuldip Paliwal, James Lyons, Abdollah Dehzangi, Alok Sharma, Jihua Wang, Abdul Sattar, Yuedong Yang, and Yaoqi Zhou. Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Scientific reports*, 5, 2015.

[46] Ronesh Sharma, Abdollah Dehzangi, James Lyons, Kuldip Paliwal, Tatsuhiko Tsunoda, and Alok Sharma. Predict gram-positive and gram-negative subcellular localization via incorporating evolutionary information and physicochemical features into chou's general pseaac. *IEEE Transactions on NanoBioscience*, 14(8):915–926, 2015.

[47] Yuedong Yang, Rhys Heffernan, Kuldip Paliwal, James Lyons, Abdollah Dehzangi, Alok Sharma, Jihua Wang, Abdul Sattar, and Yaoqi Zhou. Spider2: A package to predict secondary structure, accessible surface area, and main-chain torsional angles by deep neural networks. *Prediction of Protein Secondary Structure*, pages 55–63, 2017.

[48] Kuo-Chen Chou and Hong-Bin Shen. Recent progress in protein subcellular location prediction. *Analytical biochemistry*, 370(1):1–16, 2007.

[49] Abdollah Dehzangi, Alok Sharma, James Lyons, Kuldip K Paliwal, and Abdul Sattar. A mixture of physicochemical and evolutionary–based feature extraction approaches for protein fold recognition. *International journal of data mining and bioinformatics*, 11(1):115–138, 2014.

[50] Alok Sharma, James Lyons, Abdollah Dehzangi, and Kuldip K Paliwal. A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition. *Journal of theoretical biology*, 320:41–46, 2013.

[51] Abdollah Dehzangi, Kuldip Paliwal, James Lyons, Alok Sharma, and Abdul Sattar. A segmentation-based method to extract structural and evolutionary features for protein fold recognition. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 11(3):510–519, 2014.

[52] Abdollah Dehzangi and Abdul Sattar. Protein fold recognition using segmentation-based feature extraction model. In *Asian Conference on Intelligent Information and Database Systems*, pages 345–354. Springer, 2013.

[53] Abdollah Dehzangi, Sohrab Sohrabi, Rhys Heffernan, Alok Sharma, James Lyons, Kuldip Paliwal, and Abdul Sattar. Gram-positive and gram-negative subcellular localization using rotation forest and physicochemical-based features. *BMC bioinformatics*, 16(4):S1, 2015.

[54] Abdollah Dehzangi, Kuldip Paliwal, James Lyons, Alok Sharma, and Abdul Sattar. Enhancing protein fold prediction accuracy using evolutionary and structural features. In *IAPR International Conference on Pattern Recognition in Bioinformatics*, pages 196–207. Springer, 2013.

[55] Mark A Hall and Lloyd A Smith. Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper. In *FLAIRS conference*, volume 1999, pages 235–239, 1999.

[56] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[57] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1):389–422, 2002.

[58] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[59] Pedro Domingos and Michael Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning*, 29(2):103–130, 1997.

[60] FW Scholz. Maximum likelihood estimation. *Encyclopedia of statistical sciences*, 1985.

[61] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

[62] Yoav Freund, Robert Schapire, and N Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[63] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[64] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[65] Yan Xu, Li Li, Jun Ding, Ling-Yun Wu, Guoqin Mai, and Fengfeng Zhou. Gly-pseaac: Identifying protein lysine glycation through sequences. *Gene*, 602:1–7, 2017.

[66] Chunaram Choudhary, Chanchal Kumar, Florian Gnad, Michael L Nielsen, Michael Rehman, Tobias C Walther, Jesper V Olsen, and Matthias Mann. Lysine acetylation targets protein complexes and coregulates major cellular functions. *Science*, 325(5942):834–840, 2009.

[67] Nikolaj Blom, Thomas Sicheritz-Pontén, Ramneek Gupta, Steen Gammeltoft, and Søren Brunak. Prediction of post-translational glycosylation and phosphorylation of proteins from the amino acid sequence. *Proteomics*, 4(6):1633–1649, 2004.

[68] Niels Landwehr, Mark Hall, and Eibe Frank. Logistic model trees. *Machine Learning*, 59(1-2):161–205, 2005.

[69] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[70] Houtao Deng and George Runger. Feature selection via regularized trees. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8. IEEE, 2012.

[71] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[72] Francis Bach. Model-consistent sparse estimation through the bootstrap. *arXiv preprint arXiv:0901.3202*, 2009.