Sequence-based Predictors for Identification of Nucleosome Positioning and Transmembrane Proteins

Muhammad Tahir
13-S-AWKUM-USM-PHD-CS-02

DEPARTMENT OF COMPUTER SCIENCE,
FACULTY OF PHYSICAL AND NUMERICAL SCIENCES,
ABDUL WALI KHAN UNIVERSITY, MARDAN, PAKISTAN.
2017
Sequence-based Predictors for Identification of Nucleosome Positioning and Transmembrane Proteins

By

Muhammad Tahir

13-S-AWKUM-USM-PHD-CS-02

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

DEPARTMENT OF COMPUTER SCIENCE,
FACULTY OF PHYSICAL AND NUMERICAL SCIENCES,
ABDUL WALI KHAN UNIVERSITY, MARDAN, PAKISTAN.
2017
Author’s Declaration

I, Muhammad Tahir, hereby state that my PhD thesis titled as “sequence-based predictors for identification of nucleosome positioning and transmembrane proteins” is my own work and has not been submitted previously by me for taking any degree from this university “Abdul Wali Khan University, Mardan” or anywhere else in the country/world. At any time if my statement is found to be incorrect even after my Graduate the university has the right to withdraw my PhD degree.

Author’s Signature: ________________
Author’s Name: Muhammad Tahir
Plagiarism Undertaking

I solemnly declare that research work presented in the thesis titled “Sequence-based Predictors for Identification of Nucleosome Positioning and Transmembrane Proteins” is solely my research work with no significant contribution from any other person. Small contribution/help whenever taken has been duly acknowledged and the complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Abdul Wali Khan University, Mardan towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any materials used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of PhD degree, the university reserves the rights to withdraw/revoke my PhD degree and the HEC and the university has the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized theses.

Student/Author Signature: ____________
Name: Muhammad Tahir
Certificate of Approval

This is to certify that the research work presented in this thesis, entitled “Sequence-based Predictors for Identification of Nucleosome Positioning and Transmembrane Proteins” was conducted by Mr. Muhammad Tahir under the supervision of Dr. Maqsood Hayat. No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the Department of Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the field of Computer Science, Department of Computer Science, Abdul Wali Khan University, Mardan.

______________________________________________  _________________________________________________
External Examiner                               Supervisor
Dr. Zahoor Jan                                   Dr. Maqsood Hayat
Chairman                                         Assistant Professor
Department of Computer Science,                  Department of Computer Science,
Islamia College University, Peshawar.            Abdul Wali Khan University, Mardan.

______________________________________________  _________________________________________________
Co-Supervisor                                    Chairman
Prof. Dr. Sher Afzal Khan                        Dr. Mukhtaj Khan
Department of Computer Science,                  Department of Computer Science,
Abdul Wali Khan University, Mardan.              Abdul Wali Khan University, Mardan.

______________________________________________  _________________________________________________
Director Academics                                Dean Faculty of Physical &
Prof. Dr. Salim Ullah Khan                       Numerical Sciences
Abdul Wali Khan University, Mardan.              Prof. Dr. Aurangzeb Khan
                                                  Abdul Wali Khan University, Mardan.
Dedication

To Those I Love & Those Who Love Me
List of Publications


# Table of Contents

[Author’s Declaration] iii
[Plagiarism Undertaking] iv
[Certificate of Approval] v
[Dedication] vi
[List of Publications] vii
[Table of Contents] viii
[List of Tables] xiii
[List of Abbreviations] xiv
[Abstract] xvii

1 **Introduction** ................................................................. 1
   1.1 Deoxyribonucleic Acids .................................................. 1
   1.2 Biological Membranes and Transmembrane Proteins .......... 4
   1.3 Problem Statement ......................................................... 7
   1.4 Research Objectives and Contributions ........................... 7
   1.5 Thesis Structure .......................................................... 10

2. **Literature Survey** ........................................................... 12
   2.1 Models for Nucleosome Positioning in Genomes ............ 13
       2.1.1 Generative Models .................................................. 13
       2.1.2 Discriminative Models ............................................. 15
   2.2 Models for Transmembrane Proteins (Alpha-helices) ....... 17

3. **Implemented Approaches** ............................................... 21
   3.1 Datasets ........................................................................ 22
   3.2 Feature Extraction (Sequence Formulation Techniques) ...... 23
       3.2.1 Feature Extraction for Nucleosome Positioning ......... 23
       3.2.2 Feature Extraction for Transmembrane-Helix .......... 28
   3.3 Feature Selection Technique .......................................... 32
   3.4 Data Validation (Partition) ............................................. 34
       3.4.1 Jackknife Test ......................................................... 35
       3.4.2 Sub-sampling Test ..................................................... 35
   3.5 Learning Hypotheses ...................................................... 35
3.5.1. Support Vector Machine ................................................................. 36
3.5.2. k-Nearest Neighbor ........................................................................ 38
3.5.3. Probabilistic Neural Network .......................................................... 39
3.5.4. Fuzzy Support Vector Machine ......................................................... 42
3.6 Evaluation criteria .................................................................................. 44
3.6.1. Accuracy ........................................................................................... 45
3.6.2. Sensitivity .......................................................................................... 45
3.6.3. Specificity .......................................................................................... 45
3.6.4. Mathew’s Correlation Coefficient ....................................................... 46
3.6.5. F-measure ......................................................................................... 46

4. iNuc-STNC: A SEQUENCE-BASED PREDICTOR FOR DISCRIMINATION OF NUCLEOSOME POSITIONING IN GENOMES ................................................................. 48
4.1 Introduction ............................................................................................ 48
4.2 Materials and Methods .......................................................................... 49
4.2.1. Datasets ............................................................................................ 49
4.2.2. Feature Extraction Techniques ........................................................... 51
4.3 Results and Discussion ............................................................................ 51
4.3.1. Performance Comparison of learning hypotheses on various feature spaces using Dataset $S_1$ .................................................................................. 51
4.3.2. Performance Comparison of learning hypotheses on various feature spaces using Dataset $S_2$ .................................................................................. 53
4.3.3. Performance comparison of Learning hypotheses on various feature spaces using dataset $S_3$ .................................................................................... 56
4.3.4. Performance Comparison of iNuc-STNC model with other Models .......... 58

5. PSOFuzzySVM-TMH: IDENTIFICATION OF TRANSMEMBRANE HELIX SEGMENTS ................................................................................................. 60
5.1 Introduction ............................................................................................ 60
5.2 Materials and Methods .......................................................................... 61
5.2.1. Datasets ............................................................................................ 61
5.2.2. Feature Extraction Techniques ........................................................... 61
5.2.3. Proposed PSOFuzzySVM-TMH Prediction Model ................................. 62
5.3 Results and Discussion ............................................................................ 65
5.3.1. Performance analysis of PSOFuzzySVM-TMH on PSSM feature space .... 65
5.3.2. Performance analysis of PSOFuzzySVM-TMH on 6-letter exchange group67
5.3.3. Performance analysis of PSOFuzzySVM-TMH on Hybrid feature space .... 69
5.3.4. Performance comparison of PSOFuzzySVM-TMH model with existing models

6. CONCLUSIONS AND FUTURE DIRECTIONS .............................................. 75

6.1 Nucleosome Positioning in Genomes ................................................. 75
6.2 Transmembrane Proteins ..................................................................... 76
6.3 Future Directions ................................................................................... 76

References ..................................................................................................... 78
List of Figures

Figure 1.1 Showing the structural formulae of Purines and Pyrimidines........................................ 2
Figure 1.2 The basic structure of DNA.............................................................................................. 3
Figure 1.3 Depicts the structure of nucleosome. Each nucleosome consists of about 147 base pairs of DNA wrapped 1.67 times around a histone octamer [10]................................. 4
Figure 1.4 The Classical view of Central Dogma............................................................................... 4
Figure 1.5 Fluid mosaic model [17].................................................................................................. 5
Figure 1.6 Depicts various types of membrane proteins [21].......................................................... 6
Figure 1.7 Shows the alpha helix transmembrane protein................................................................. 7
Figure 1.8 Framework of Research work.......................................................................................... 9
Figure 2.1 Number of entries in UniProtKB/Swiss-Port database.................................................. 12
Figure 2.2 A Hidden Markov Model ................................................................................................ 14
Figure 3.1 Structure of Machine Learning processes....................................................................... 22
Figure 3.2 Shows the procedure of DNC ......................................................................................... 26
Figure 3.3 Shows the process of TNC............................................................................................... 27
Figure 3.4 Support Vector Machine.................................................................................................. 38
Figure 3.5 Example of K-Nearest Neighbor Algorithm................................................................. 39
Figure 3.6 Architecture of probabilistic neural network............................................................... 41
Figure 4.1 Framework of iNuc-STNC Model ..................................................................................... 49
Figure 4.2 Performance of various learning hypothesis and feature spaces on dataset S_1............. 53
Figure 4.3 Performance of various learning hypothesis and feature spaces on dataset S_2............. 55
Figure 4.4 Performance of various learning hypothesis and feature spaces on dataset S_3............. 57
Figure 5.1 The framework of the proposed prediction PSOFuzzySVM-TMH model.................. 63
Figure 5.2 Performance of PSSM feature space for low resolution dataset................................. 66
Figure 5.3 Performance of PSSM feature space for High resolution dataset .................. 67

Figure 5.4 Performance of 6-letter exchange group feature spaces for low resolution dataset 68

Figure 5.5 Performance of 6-letter exchange group feature spaces for High resolution dataset .................................................................................................................. 69

Figure 5.6 Performance of Hybrid feature spaces for low resolution dataset............... 71

Figure 5.7 Performance of Hybrid feature spaces for high resolution dataset............... 71
List of Tables

Table 3.1 Categorization of 20 Amino Acids ................................................................. 32
Table 3.2 A Confusion Matrix ......................................................................................... 45
Table 4.1 Resources for Benchmark Datasets ................................................................. 51
Table 4.2 Performance Analysis of various Feature Spaces using $S_1$ ......................... 52
Table 4.3 Performance Analysis of various Feature Spaces using $S_2$ .......................... 55
Table 4.4 Performance analysis of various feature spaces using $S_3$ ......................... 57
Table 4.5 Comparison of our proposed iNuc-STNC model with other models ............ 59
Table 5.1 Performance analysis of PSSM feature space at different levels ................... 66
Table 5.2 Performance analysis of 6-letter exchange group feature spaces at different levels 68
Table 5.3 Performance analysis of Hybrid feature space at different levels .................. 70
Table 5.4 Performance comparison of PSOFuzzySVM-TMH model with existing models... 73
### List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Adenine</td>
</tr>
<tr>
<td>AA</td>
<td>Amino Acid</td>
</tr>
<tr>
<td>bp</td>
<td>base pair</td>
</tr>
<tr>
<td>C</td>
<td>Cytosine</td>
</tr>
<tr>
<td>DNA</td>
<td>Deoxyribonucleic Acids</td>
</tr>
<tr>
<td>DNC</td>
<td>Dinucleotide Composition</td>
</tr>
<tr>
<td>FP</td>
<td>False Positive</td>
</tr>
<tr>
<td>FN</td>
<td>False Negative</td>
</tr>
<tr>
<td>G</td>
<td>Guanine</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Model</td>
</tr>
<tr>
<td>KNN</td>
<td>K-Nearest Neighbor</td>
</tr>
<tr>
<td>MCC</td>
<td>Mathew’s Correlation Coefficient</td>
</tr>
<tr>
<td>NAC</td>
<td>Nucleic Acid Composition</td>
</tr>
<tr>
<td>PSO</td>
<td>Particle Swarm Optimization</td>
</tr>
<tr>
<td>PNN</td>
<td>Probabilistic Neural Network</td>
</tr>
<tr>
<td>PseAA</td>
<td>Pseudo Amino Acid</td>
</tr>
<tr>
<td>PSSM</td>
<td>Position Specific Scoring Matrix</td>
</tr>
<tr>
<td>PSI-BLAST</td>
<td>Position Specific Iterated BLAST</td>
</tr>
<tr>
<td>RNA</td>
<td>Ribonucleic Acid</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>T</td>
<td>Thymine</td>
</tr>
<tr>
<td>TP</td>
<td>True Positive</td>
</tr>
<tr>
<td>TN</td>
<td>True Negative</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>TM</td>
<td>Transmembrane</td>
</tr>
<tr>
<td>TMH</td>
<td>Transmembrane Helix</td>
</tr>
<tr>
<td>TNC</td>
<td>Trinucleotide Composition</td>
</tr>
<tr>
<td>STNC</td>
<td>Split Trinucleotide Composition</td>
</tr>
<tr>
<td>NMR</td>
<td>Nuclear Magnetic Resonance</td>
</tr>
</tbody>
</table>
Acknowledgements

First of all, I am very thankful to Almighty Allah, for His divine guidance and providence. His support, goodness, kindness, and blessings are always with me. He provided me the strength, determination and knowledge to accomplish my PhD research work. It is the special blessing of Almighty Allah to have a community of sincere teachers, praying parents and cooperative colleagues and friends who all helped me in making my endeavors a success.

Every PhD student dreams for a visionary supervisor. I am extremely thankful to Allah, for connecting me up with the best possible supervisor Dr. Maqsood Hayat. His guidance, inspirations and motivations enabled me to bring out my best. I learnt many things from him, and the list is quite long but in short, he introduced me to the scientific writing, guided me to the art of critical thinking and patience throughout my research work at Abdul Wali Khan University, Mardan. I pay my indebtedness to him and his greatness.

I am very thankful to my co-supervisor Dr. Sher Afzal Khan for his kind support and guidance. I would also like to pay my gratitude to chairman department of computer science Dr. Mukhtaj Khan for his guidance, advice, and valuable comments during my PhD. I would also appreciate my friends and colleagues for their cooperative and encouraging behavior during my study at Abdul Wali Khan University, Mardan.

I would certainly like to express my deepest gratitude for my loving parents, brothers, sisters and family whose prayers and support, throughout my studies, have made all this possible. Without their moral support and encouraging behavior, the completion of this research work would not have been possible. To my lovely daughter, Bareera Tahir, thank you for being cute, making surprises in my and bringing happiness to the family.

Finally, and most importantly, I would like to thank my wife, for her consistent encouragement and support in my research and for all the wonderful moments we shared together.

Muhammad Tahir
Abstract

In this thesis, the research work was carried out in two phases. In Phase-I, an intelligent computational model is developed for identification of nucleosome positioning in genome. Nucleosome positions perform a distinguished role in regulating the genes activities, due to which it was targeted in phase-I. Nucleosome is a vital reiterating unit of eukaryotic chromatin, contains DNA enclosed around a histone core. The nucleosomes restrict the accessibility of the enclosed DNA to transcription factors and other DNA-binding proteins. Owing to the promising role of nucleosomes, a more accurate and an efficient intelligent automated model iNuc-STNC has been developed. In this model, three different feature extraction techniques including dinucleotide, trinucleotide and split trinucleotide compositions were adopted in order to excerpt prominent, salient and high variant numerical descriptors. Various learning hypotheses such as k-nearest neighbor, probabilistic neural network, and support vector machine were utilized for classification. The predictive outcomes of iNuc-STNC model were encouraging and remarkable than the existing approaches so far in the literature. It is thus highly observed that the developed method will be more helpful and expedient for basic academic research and pharmaceutical industry in the designing of drug.

In Phase-II, an automated model is developed for discrimination of transmembrane protein structures. Transmembrane proteins manage various intra or extra cellular processes of a cell. In addition, it performs signaling, cell recognition, cell adhesion, and cell to cell interaction. However, all the essential clues regarding the functions and structures of transmembrane proteins are reflected from transmembrane topology. Owing to the limited number of recognized structures, it is very hard to truly reflect the target proteins. In this regards, a computational prediction PSOFuzzySVM-TMH model was developed to correctly identify the location of transmembrane helix from their primary sequences. The protein sequences were numerically expressed by two various feature extraction techniques such as 6-letter exchange group representation and position specific scoring matrix in order to exploit all the salient, pronounced and variant numerical descriptors. Further, evolutionary feature selection technique namely: particle swarm optimization was used to condense the feature space by eradicating the irrelevant and noisy information in order to enhance the learning and generalization capability of predictive model. The Fuzzy SVM is used as learning
hypothesis for classification. 10-fold cross validation test is applied for the assessment of PSOFuzzySVM-TMH model at different levels i.e., per segment, per protein, and per residue, using two different benchmark datasets. After experimental analysis, it is realized that the PSOFuzzySVM-TMH model has identified the transmembrane proteins at different levels with high true classification rates than the existing models so far. All these achievements are credited to by incorporating the concept of fuzzy with SVM, evolutionary selection of high variant features, and clearly discerning the motif of target classes.
Chapter 1

Introduction

All living organisms consist of cells, which may be unicellular or multi cellular. The cell performs different functions such as reproduction, energy conversion, molecular transportation, and identity maintenance. It is a fundamental unit of living organisms, which have cellular organelles such as, Nucleus, Golgi complex, Mitochondria, Endoplasmic reticulum and Ribosomes. Among these cellular organelles, nucleus is a membrane-bound organelle, having super coiled deoxyribonucleic acid (DNA) molecule, which has hereditary materials that transfer characteristics from parents to offspring or from one generation to next generation.

1.1 Deoxyribonucleic Acids

DNA is a longer polymer of nucleic acids having macro-molecules, which consist of nucleotides. These nucleotides are made of deoxyribose (sugar), nitrogenous base, and phosphate group. The hereditary information is stored on the basis of nucleotide chain, which consists of adenine (A), guanine (G), cytosine(C), and thymine (T). The ‘A’ and ‘G’ are called purine whereas ‘C’ and ‘T’ are known as pyrimidine. They have a specific property of bonding for the formation of double helical structure of DNA. Purines are double ring structure and pyrimidines are single ring structure [1]. In the generic structure of DNA, purines always make hydrogen bonding with the pyrimidines i.e., ‘A’ makes a double hydrogen bonding with ‘T’ and ‘C’ forms a triple hydrogen bonding with ‘G’. In Figure 1.1, the structural formulae of Purines and Pyrimidines are shown.
In double helical structure of DNA, there are two strands. One strand is called template strand, also known as coding strand, three to five prime (3’-5’). The other strand is called non-template strand, also known as non-coding strand, five to three prime (5’-3’). The backbone of the helical structure of DNA consists of sugar-phosphate, linked together with phosphodiester bonds. The distance per turn is 3.4 nm, distance between two nucleotides is 0.34 nm, distance between two anti-parallel strands of DNA is 2 nm, and the number of nucleotides per turn is 10. The DNA helix consists of major and minor grooves, the anti-parallel backbone strands are close together in some places, known as the major groove, while in some places, they are far away, known as the minor groove, as shown in Figure.1.2.
DNA constitutes in a super coiling structure called as chromatin. Nucleosome is the primary unit of eukaryotic chromatin, which consists of histone proteins and DNA molecules [2]. The core histone proteins consist of four sub-units such as, H2A, H2B, H3 and H4, whereas, the linker histone is H1. These proteins are enriched with basic amino acids such as arginine and lysine [1, 3].

Double helical DNA strand around the core histone particles in a 145–147 bps nucleotides long is a left-handed super helix [4, 5]. This DNA is actually divided into two parts including a core DNA and a linker DNA. Through the linker DNA, the short DNA sequences are connected to the adjacent nucleosome, which ranges from 10-100 bp [6-8]. The adjacent nucleosomes are associated with each other through linker DNA and short DNA sequences, which range from 20-60 bp [9, 10] as illustrated in Figure 1.3. The final length of the DNA in nucleosome is 166–167 bp, which is two full turns [11] and this unit is called chromatosome. The packaging of DNA around the histone octamer, performs vital roles in biological processes such as DNA replication, RNA splicing, transcriptional control, and DNA repair mechanisms [12-14].
Figure 1.3 Depicts the structure of nucleosome. Each nucleosome consists of about 147 base pairs of DNA wrapped 1.67 times around a histone octamer [10].

The molecular central dogma explains the transformation of genetic information from DNA to RNA and RNA to protein. The DNA is transcribed into a specific messenger RNA (mRNA). Each specific mRNA has information for a specific protein, which synthesizes a particular protein [15, 16]. Figure 1.4 shows the classical view of a central dogma.

Figure 1.4 The Classical view of Central Dogma

1.2 Biological Membranes and Transmembrane Proteins

Cell membrane is a thin layer that covers the external boundary of a cell where the external boundary is known as plasma membrane. Biological membrane performs vital functions in a cell such as signaling, barrier, energy conversion, recognition, and cell
subdivision [17]. The biological membrane is mostly composed of a lipid bilayer proteins and carbohydrate [18]. These proteins are the main components of a cell called membranous protein, which performs a major role in cellular process ranging from simple transport to sophisticated signaling pathways. Membrane proteins consist of one or more transmembrane helices mainly composed of α-helices, which are responsible for the orientation of membrane proteins to lipid bilayer. In addition, significant biological processes such as signaling, cell recognition, cell adhesion and cell-to-cell interactions are performed by membrane proteins. The transmembrane proteins are the sub-part of integral proteins. They have three parts: one part is located inside the cell, the second part resides in the lipid bilayer, and the final part is exterior to the cell [19]. Another important component of protein structure is β barrel/strand.

![Image of Plasma Membrane Structural Components](image.png)

**Figure 1.5 Fluid mosaic model [17]**

Alpha-helical transmembrane proteins constitute in every cell membranes including external membranes as well as various functions such as light driven transporters, enzymes, electrochemical potential-driven transporters, and so forth [20]. Alpha-helix transmembrane proteins contain a chain of hydrophobic amino acids, which are linked to transmembrane helices by extra membranous loop region. Single
pass alpha helix transmembrane can further be classified into different sub-classes; single-pass type-I transmembrane protein, single-pass type-II transmembrane protein, single-pass type-III transmembrane protein and single-pass type-IV transmembrane protein as illustrated in Figure 1.6 as well.

![Image of membrane proteins](image)

**Figure 1.6 Depicts various types of membrane proteins [21]**

Single pass Type-I transmembrane protein presents its extracellular on N-terminus and cytoplasmic on C-terminus whereas in single pass type-II transmembrane protein C-terminus is present on exoplasmic side and N-terminus on cytoplasmic side having a small number of cleavable endoplasmic reticulum signal sequence [21]. In each type, the polypeptides consist of about 25 hydrophobic amino acids, cross the lipid bilayer only once. Similarly, multi-pass proteins are classified in to two different parts; alpha-helix multi-pass and beta-barrels multi-pass. Figure 1.7 shows the structure of alpha-helix transmembrane protein. Alpha-helix multi-pass transmembrane protein also known as tetraspanins, the polypeptides pass the lipid bilayer several times mostly four to seven times. However, C- and N- termini remain on the same side of membrane in case of even number of alpha-helices in multi-pass transmembrane protein [22].
1.3 Problem Statement

Identification and annotation of genome and protein sequences due to their rapid exploration in the form of huge and unprocessed data is becoming a challenging task in computational biology, genomics, and bioinformatics. In this regards, various conventional methods have been investigated. Although, conventional methods have obtained some considerable results; they are almost impossible sometimes for proteomic and microscopic detection of some of the species. It is mainly due to the complex structure and lack of availability of recognized number of proteins. Therefore, it is really a challenging task to develop an effective and high throughput automated model for recognition of uncharacterized biological molecules.

1.4 Research Objectives and Contributions

The genome level DNA and protein sequences are regularly increased in database due to fast technological improvement in biological systems. Early in 1986, 3,939 protein sequences were found in Swiss-Prot database according to statistics released [21]. Whereas according to the recent release of 18th January 2017 UniProtKB/Swiss-Prot holds 553,474 protein sequences, which reveal that 140 times increase has been found as compared to the early report. The discrimination and annotation of these unprocessed data are the main challenges in the area of computational biology and bioinformatics.
Laboratory experimental approaches provided encouraging results; but due to the lack of proteins structure information and vagueness of available motifs, proteomics, and microscopic detection for some species, these techniques are almost impossible to be applied. Owing to this reason, the experimental approaches are being applied on a limited number of genomes and proteomes. Therefore, for identification of uncharacterized proteins, the demand for reliable, automatic, and fast computational models are increasing day by day.

The main objective of our research work is to identify nucleosome positioning in genomes and transmembrane proteins using machine learning and pattern recognition contemporary approaches. The amino acids sequence contain different patterns (motifs) and useful hidden information (primary, secondary, evolutionary and etc). This concealed and reliable information provide useful clue for identification of transmembrane proteins and nucleosome position in chromatin. However, the challenging task is how to extract these salient features from topogenic sequences. In this connection, such a feature extraction strategy or technique is needed, which not only extracts numerical values from protein sequences but also play a significant role in identification of transmembrane proteins and nucleosome positioning. A number of researchers have developed various prediction systems for identification of nucleosomes and transmembrane proteins. Despite tremendous improvement has been observed through pattern recognition and machine learning based methods; the room is still available for more consideration and exploration. Therefore, development of a novel computational technique as well as enhancing the performance of existing techniques for prediction of nucleosomes and transmembrane proteins will be considered in this work.

The major objectives of the proposed system are:

- To develop an automatic, fast and robust model using machine learning and contemporary intelligence techniques.

- To obtain high success rates compared to the existing models in terms of prediction accuracy.

- To develop a web predictor, which would be useful for drugs design, proteomics, and academic research.

Our research work was carried out in two phases, which are illustrated in Figure 1.8.
Phase-I: Nucleosome positioning prediction in chromatin (genomes) material.

- The genome sequences were expressed using three feature extraction techniques including dinucleotide composition (DNC), trinucleotide composition (TNC), and split trinucleotide composition (STNC).

- Different learning hypotheses were employed for classifications namely: k-nearest neighbor (KNN), probabilistic neural network (PNN), support vector machine (SVM) and quantitative score was computed using metrics including accuracy, specificity, sensitivity, and Mathew’s Correlation Coefficient (MCC). Whereas these metrics were examined by rigorous statistical cross validation tests namely jackknife tests.
Phase-II: Alpha helices prediction in membrane protein.
Membrane protein structures have been identified as alpha helix transmembrane proteins

- 6-Letter exchange group representation and position specific scoring matrix (PSSM) were used for feature extraction.

- In order to avoid unnecessary and irrelevant features, features selection method such as Particle swarm optimization (PSO) was applied to reduce the computational cost and enhance performance and speed.

- Fuzzy support vector machine was employed as learning hypothesis.

- 10-fold cross validation test was applied to investigate the generalization power of learning hypothesis.

- The performance was measured in terms of per residue, per protein and per segment levels.

1.5 Thesis Structure

In Chapter 2, literature survey regarding nucleosomes and transmembrane proteins is discussed. The literature survey exhibited that various traditional and computational models have been developed by many researchers for identification of nucleosome positioning in chromatin material and transmembrane proteins.

In Chapter 3, implemented approaches of our research work are discussed in detail. According to machine learning, the processes are presented as database, feature extraction techniques, feature selection techniques, data validation, learning hypotheses, and assessment criteria.

The first phase of research was presented in Chapter 4, which demonstrates contribution regarding nucleosome positioning in genomes. In this chapter, we have suggested a predictive computational model for nucleosome positioning in genomes. The model was designed on the basis of DNC, TNC, and STNC as feature extraction scheme. SVM, PNN, and KNN were utilized as learning hypotheses. The predictive quality was computed using various metrics including accuracy, specificity, sensitivity,
and MCC. Whereas these metrics were evaluated by rigorous statistical cross validation tests namely jackknife tests.

The second phase of research about transmembrane protein structure was discussed in chapter 5. Physicochemical properties of amino acids and evolutionary profiles position specific scoring matrix (PSSM) were used as feature extraction techniques. Particle swarm optimization (PSO) was employed as features selection technique whereas fuzzy support vector machine was applied as learning hypothesis. 10-fold cross validation test was employed to examine the performance of learning hypothesis. The performance was investigated in terms of per residue, per segments, and per protein levels.

Chapter 6 concludes the thesis work and draws a conclusion. Major achievements have been discussed in this chapter. Some aspects are also highlighted that still need more future consideration.
Chapter 2

LITERATURE SURVEY

The identification of proteins and DNA was carried out through conventional experimental methods like Nuclear Magnetic Resonance (NMR), filter binding assays, and X-ray crystallography [23-28]. Owing to a limited number of genome and proteomic structures availability, the experimental methods were restricted. In addition, the other major issues reported in conventional methods were expensive w.r.t time and cost, and lack of laboratory equipment. Besides, due to the fast technological improvement in biological systems, a huge growth has been noticed in databases of genomes and protein sequences. For instance, in early 1986, the statistics report showed that the protein sequences in UniProtKB/Swiss-Prot were 3,939 [21, 29, 30]. Whereas according to the recent release of 18th January 2017 UniProtKB/Swiss-Prot holds 553,474 protein sequences, which reveal that 140 times growth has been observed compared to the early report. The growth of the database is illustrated in Figure 2.1.

![Figure 2.1 Number of entries in UniProtKB/Swiss-Port database](image-url)
Due to the huge exploration of biological sequences, the recognition and classification of these unprocessed data are the challenging jobs in the area of bioinformatics and proteomics. Owing to lots of issues in traditional approaches, the researchers divert their attention towards the computational methods by applying statistical and machine learning methods. In these methods, prediction was carried out by profiled based, structure-based and sequence-based methods. Among these methods, the sequence-based method was considered quite auspicious [28].

After achieving some considerable results, the computational approaches became the focusing area of investigators and demands for more accurate, reliable, automatic, and fast computational models. The performance of statistical and machine learning based models has been analyzed by using various performance metrics such as sensitivity, specificity, accuracy, recall, precision, and Mathew’s Correlation Coefficient. Everyone endeavored to improve the predictive quality of their model with respect to accuracy. Therefore, in the area of nucleosome positioning in genomes and transmembrane proteins, several contemporary machine learning and evolutionary computing methods were carried out for the development of computational prediction model. This chapter presents a brief review about literature survey regarding nucleosome positioning in genomes and transmembrane proteins.

2.1 Models for Nucleosome Positioning in Genomes

Numerous studies were performed for prediction of nucleosome positioning in genomes [10, 31-38]. Some of them have predicted nucleosomes with quite accurate results by placing isolated nucleosomes. In this study, we have categorized these models into two major groups: generative and discriminative models.

2.1.1. Generative Models

In machine learning, generative models attempt to capture fundamental patterns from the provided data. The generative models can reflect intrinsic hidden connections, which effect the observed data distributions and yield good data representations that are more considerable as input for machine to manage it. Generative models are broadly applied in the fields of computer vision as well as in computational biology. Hidden Markov Model (HMM) is a simplest and more applicable model of generative models [39].
In Figure 2.2, nodes or states $S_i; i=1,2,...,n$ denote the hidden states of the system and state $b_t$ symbolizes the observed state at each time step. The arrows symbol demonstrates transitions, which are chosen by a transition probability distribution.

In a Markov model, the system moves from one state to another state on the basis of specified probability, which is based on previous state values.

$$P(b_{t+1} = S_j | b_t = S_i, b_{t-1} = S_k,...)$$  \hspace{1cm} (2.1)

where $S_i$ is hidden states, $b_t$ is observed state at time $t$. First-order Markov model for special case is, time $t+1$ state depends only at time $t$ state, regardless previous time states.

$$P(b_{t+1} = S_j | b_t = S_i, b_{t-1} = S_k,...) = P(b_{t+1} = S_j | b_t = S_i)$$  \hspace{1cm} (2.2)

Suppose the transition probabilities are time independent, then the transition probabilities will be

$$a_{ij} = P(b_{t+1} = S_j | b_t = S_i)$$  \hspace{1cm} (2.3)

$O = B = b_1b_2...b_T$ is the state observation sequence represented by $O$. Its probability can be computed as:

$$P(O = B | A, II) = P(b_1) \prod_{t=2}^{T} P(b_t | b_{t-1}) = \pi_1 a_{b_1b_2} a_{b_2b_3}...a_{b_Tb_1}$$  \hspace{1cm} (2.4)
where the probability of the first state $b_1$ is $\pi_{b_1}$, the probability of the state going form $b_1$ to $b_2$ is $a_{b_1b_2}$ and so on. Finally, these probabilities are multiplied in order to get the probability of entire sequence [2, 40, 41].

Generative models have been studied by a number of researchers. Segal et al., developed a probabilistic model by computing nucleotides probabilities and higher order dependencies among nucleotides. By implementing the steric impediment impacts between nucleosomes, they gathered a set of non-overlapping nucleosome sequences. The HMM was found out for measurable nucleosome positioning from a training data of adjusted nucleosome-bound sequences. The trained model was utilized to identify non-overlapping nucleosome positions measurably along a given chromosome utilizing dynamic programming. Experimental results demonstrated that the HMM model locates 54% of the nucleosomes inside 35 bps of their actual positions, which is 15% more than what might be expected of random placement of a same number of nucleosomes in the genome [2].

Likewise, Kaplan et al., and Field et al., used various k-mer methods for enhancing the performance of the developed models [33, 42]. Besides, for position-dependent probability the global information of length 5 base pair was also used with frequency base features. The result demonstrates that the intrinsic DNA sequence preferences of nucleosomes perform the key role in representing nucleosome organization [33, 42]. Similarly, Xi et al., proposed a novel duration Hidden Markov Model (dHMM) to collect nucleosome positioning information by implementing the linker DNA length as well as nucleosome positions. It is observed that the proposed kernel method is considered nonparametric and robust by updating the linker length distribution iteratively in order to enhance the performance measure such as sensitivity and also to minimize false discovery rate (FDR) in prediction [37].

2.1.2. Discriminative Models

Discriminative models are also applied in different application areas of bioinformatics and computational biology. Discriminative models use the conditional probability distribution function $P(y|x)$, where the generated output $(y)$ is fully dependent on the provided input $(x)$. In addition, discriminative models do not produce instances from the joint distribution of $x$ and $y$ like other models. In regression and classification, the distributive models obtained efficient results in such cases where there is no need of
joint distribution. Various computational models have been developed based on the **k-mer features** of nucleosome positioning sequence in genomes. The k-mer feature extraction technique is considered the most discriminative technique in which the frequencies among the nucleotides (A, C, G and T) are computed where \( k \) value is selected from 1 to 6. For a 50 bp DNA sub-sequence, each k-mer value and its reverse complements are considered to be the same (e.g. "AA", "AC", "AG" and "AT" correspond to the same 2-mer feature). After calculating the frequency for each value of \( k \), the corresponding output is then normalized.

Satchwell et al., for the first time observed the periodical occurrence of nucleotides in particular order such as two residues pair is known as dinucleotide (AA, AC, … TT) and three residues pair is known as trinucleotide (AAA, AAT, …, TTT) by examining 177 examples of nucleosome-bound sequences from the chicken erythrocytes [43]. Furthermore, Peckham et al., introduced the SVM based model, used sequence-based features to analyze some oligo-nucleotides implicated in nucleosome formation and exclusion. In order to distinguish forming and inhibiting nucleosome sequences with high precision, they have applied the frequencies of the k-mers [31, 41]. Further, Gupta et al., developed a method for accurately predicting human nucleosome positions by using sequence-based feature space in order to train the model effectively [32]. Furthermore, Yuan and Liu, introduced a sequence based novel algorithm, which is also known as N-Score for identification of nucleosome positioning in yeast. Wavelet transformation method was used to extract dinucleotide features, considered all the linker DNA sequence information and long-range sequence information. Logistic regression model was utilized to clearly identify the signals, which are useful for distinguishing linker and nucleosome sequences [44].

Gou et al., in 2014 developed an ‘iNuc-PseKNC’ method for the prediction of nucleosome positioning in genomes. They have highlighted the issue of small dataset in their work. Guo et al., constructed the stringent benchmark datasets of nucleosome-forming and nucleosome -inhibiting sequences with low similarities [10]. The performance of ‘iNuc-PseKNC’ predictor demonstrated good results compared to previous works. The DNA sequences were represented by a new concept known as pseudo k-tuple nucleotide composition in which six DNA local structural physicochemical properties were utilized. Support vector machine was used as a learning hypothesis for prediction of nucleosome positioning in genomes [10].
Other various computational models have been developed where the notion of pseudo amino acid (PseAA) composition was extensively employed. The general concept of PseAA composition was extended for DNA representation and developed various models such as repDNA [45], Pse-in-One [46], and iDNA-KACC [47]. In addition, some predictors namely: iRSpot-EL [48] and iDHS-EL [49] were also developed by Liu et al. The idea of PseKNC was effectively adopted and demonstrated in DNA/ RNA such as predicting recombination spots [50-52], identifying nucleosome [50], predicting splicing site, identifying translation initiation site [53], predicting promoters [54], identifying RNA and DNA modification [55, 56], and identifying origin of replication [57].

2.2 Models for Transmembrane Proteins (Alpha-helices)

The functionalities of transmembrane proteins depend upon the formation of its structure and topology. Therefore, it is inevitable to examine the spatial organization of transmembrane protein. However, knowledge regarding transmembrane proteins reflects some valuable evidence in expressing their structure and topology. On the basis of these clues, it can be very easy to determine the segments, penetrating within membrane and loops on either side. Initially, some visualization techniques were carried out for identification of transmembrane proteins namely: “helical wheel” [58] and “helical net” [59]. Furthermore, a quantitative method “hydrophobic moment” [60] was also introduced in 1982. These methods have only focused on physiochemical behaviors of amino acids because the regions of alpha-helix have already been identified.

A number of researchers have used biochemical and spectroscopic experiments like solid state Nuclear Magnetic Resonance (NMR) [61-63], infrared spectroscopy [64, 65], and electron microscopy [66, 67]. The major issues reported in experimental approaches are shortage of raw materials for toxicity, crystallization, inclusion bodies, and so on. In addition, they are only targeting the structural knowledge of low-resolution on membrane proteins, which is very tedious and time consuming. Due to these problems, very limited number of membrane protein structures are identified. These problems were minimized by considering the topology of membrane proteins as an alternative. Topology determines the transmembrane segments, which are embedded within membrane and the side of membrane, which links the C-terminus with N-
terminus by loop [68]. These kinds of information are very essential for both functional and structural classification of proteins. The length of transmembrane protein segments is almost 25 hydrophobic residues but sometimes, it can be between 15-40 residues long [69]. On the other hand, turns are formed in the hydrophobic loops due to the presence of glycines and pralines amino acids [70]. Kyte and Doolittle have utilized the hydrophobicity scale for identification of transmembrane segments by applying 19 residues long sliding window and determined the length of transmembrane segments around 20 to 30 residues [71, 72]. Furthermore, Hydrophobicity scales, amino acid propensities and window sizes have been optimized to improve the quality of predictors [18, 73, 74]. A ‘positive inside rule’ is established in that rule, more positive charge amino acids are detected, in short cytoplasmic loops, which link transmembrane segments to extracellular loops [75]. Other physicochemical properties of amino acids like non-polar phase helicity [76], charges [77, 78], multiple sequences alignment [79, 80], TOP-Pred [81], DAS-TMfilter [82], and SOSUI [78] were utilized for the identification of transmembrane segments and topology.

In a sequel, Hayat and Khan have developed a prediction model WRF-TMH for successful identification of TMH segments. In the WRF-TMH model, physicochemical properties of amino acids and composition index were used for feature extraction, whereas weighted random forest is used as learning hypothesis [83]. Likewise, Deng et al., have developed an evidential reasoning base model TOPPER, in which the success rates of various individual learning algorithms were converted into basic probability assignments according to the confusion matrix [84]. Shen and Chou introduced a new model known as MemBrain predictor; it is a sequence-based analysis tool for structural and functional characterization of helical membrane proteins, which combines several modern bioinformatics approaches [81].

These methods have identified transmembrane segments with encouraging results, however, in case of topology, the results were not satisfactory. Furthermore, computational methods have utilized to predict membrane topology more accurately. In addition, different investigators have applied various statistical and operation engines including Hidden Markov Models (HMMs) [85, 86], support vector machine (SVMs) [87], and artificial neural networks (NNs) [87] for identification of transmembrane helix segments.

Simon and Tusnady, have introduced the concept of HMMs for prediction of transmembrane topology and developed a model HMMTOP [85, 88, 89]. Further,
Krogh et al., brought efforts and developed TMHMM model for discrimination of transmembrane topology with high confidence. TMHMM constructs a cyclic model for a transmembrane helix with seven states while HMMTOP model utilized HMMs to differentiate among five structural states such as helix core, helix caps inside and outside loop, and globular domains. The states are connected with each other via transition probabilities. Despite these efforts, multiple sequences alignment, and computational cost still remain the target issues. In addition, the main issue reported in HMMs based models is the execution of short residues segments (less than 16) and long residues segments (more than 35) [90].

Likewise HMMs, artificial neural networks were also applied for handling the problems of bioinformatics. The simplest type of neural network is feed-forward neural network. In this model, information moves from input states or nodes to output states through any hidden states. However, the information moves in one direction subsequently no cycles or loop in the network will occur. Rost et al., developed PHDhtm model, whereas Jones introduced MEMSAT3 model for prediction of transmembrane helix based on neural network [79]. The former model PHDhtm used multiple sequences alignment approach and combination of two feed-forward neural networks to execute a consensus identification of transmembrane helix. The first feed-forward neural network establishes a ‘sequence-to-structure’ network, which demonstrates the structural propensity of the central residue in a window. The second feed-forward neural network constructs a ‘structure-to-structure’ network. After that, the propensities are smoothed before applying positive-inside rule to generate overall topology. The later model MEMSAT3 utilized dynamic programming in conjunction with feed-forward neural network. This model not only identified transmembrane helix but also scored the topology to predict possible peptides signal [91].

Besides, Yuan et al., and Lo et al., applied SVM for the prediction of transmembrane protein topology. Initially, multiclass issue was raised in SVM because it was developed for binary classification [92, 93]. In contrast, HMMs and neural network based models are suitable of handling multiclass problems. Multiclass ranking SVMs are available, but they are usually inapplicable in many problems because there is no such mathematical function available to easily discriminate the classes from one another [94]. SVMs have the ability to learn intricate relationships among various amino acids in a peptide by which they are trained. SVM is flexible to the over fitting problem compared to other learning algorithms. However,
many adjustable parameters for optimization solution may cause huge time consumption.

Additionally, for the assistance of researchers and academies various user-friendly web predictors have been developed. some of them are presented here: SVMtop [95], TopPred [96], PHDhtm [97], HMMTOP [98, 99], TMHMM [100, 101], MEMSAT [102], TMMOD [103], Phobius [104], ENSEMBLE [105], PONGO [106], PRODIV_TMHMM [107], HMM-TM [108], MemBrain [109], MEMPACK [110], and MEMSAT-SVM [111]. Several studies have focused on accuracy whereas some have emphasized on specificity and sensitivity for evaluating their developed models [109, 112, 113]. Moreover, few researchers have highlighted only reliabilities and sensitivity rather than accuracy [114-118].
Chapter 3

IMPLEMENTED APPROACHES

In the previous chapters, we have discussed literature review comprehensively. In this chapter, we will explain the implemented approaches in this research. Machine learning is the study of constructing a system that can learn from the environment, observations and past experience. It is broadly classified into three categories: supervised learning, unsupervised learning and semi-supervised learning. In supervised learning, actual information about the learning data is provided, whereas, in unsupervised learning, no target information is available about the data. The semi-supervised learning utilizes both unlabeled and labeled data to perform an otherwise unsupervised learning or supervised learning task [119-121].

According to broad analyses carried out in numerous research studies [10, 122-127], a reliable, accurate and fast computational model will be developed by accomplishing Chou’s 5-steps procedure [128]. Various machine learning processes are depicted in Figure 3.1.
In the following subsections, we will briefly discuss different Machine Learning processes involved in Chou's 5-step procedure.

### 3.1 Datasets

For the development of a stochastic based computational model, it is necessary to create or select a consistent and standard dataset for training and testing the model. However, in case of extraneous and erroneous dataset, subsequently, the success rate of prediction model must be inconsistent and unreliable. Looking at the significance of dataset, five various datasets are used in this study. In the first phase of our research, three distinct benchmark datasets are utilized for nucleosome positioning in genomes. These datasets contain DNA sequences of variable lengths. DNA sequence is the polymer of four nucleotides: adenine, guanine, cytosine, and thymine. They must be in FASTA format (>) as shown below.
In the second phase of our research, two different benchmark datasets for TM proteins are used. These datasets are composed of protein sequences of variable lengths. Protein sequence is the chain of twenty distinct amino acids. Similar to DNA sequence, the sequence of a protein also needs to be in FASTA format (>) as presented below.

>41BB_HUMAN Q07011
MGNSCYNIVATLLLLVLFERTSLQDSPCNCPAGTFCDNNRNQICSPCPNSFS
SAGGQRTCDICROCKVFRTRKECSSTSNACEDCCTPGFHCLGAGCSMCEQDC
KQGQELTKGCKDCCFGTFNDFQKRGIICRPWTNCSDLGKSVLVNGTKERDVV
CGPSPDLSPGASSVTPPAPAREPGHSPQIISSF

3.2 Feature Extraction (Sequence Formulation Techniques)

The second process of machine learning is to elicit numerical values from DNA/Protein sequences because the classification algorithms require numerical attributes for learning. In addition, these values discern the specific pattern or motif of the target class, which make it easy for classification algorithm to correctly predict the target class. In this work, various sequence formulation techniques have been utilized to extract prominent, salient, and high varied numerical descriptors from DNA/Protein sequences. A detailed discussion of these techniques is given below:

3.2.1. Feature Extraction for Nucleosome Positioning

This section presents those feature extraction techniques that are applied for nucleosome positioning in genomes. Suppose $S$ is an $L$ nucleic acid residues long DNA sequence as shown in Equation 3.1.

$$S = N_1N_2N_3N_4N_5N_6N_7...N_L$$  (3.1)
where \( N_1, N_2, N_3 \) represent the residues at positions first, second and third nucleotides, and \( N_L \) represents the last residue at position \( L \) in a DNA sequence [129-131]. These nucleotides are

\[
N_i \in \{ \text{A (adenine), C (cytosine), G (guanine), T (thymine)} \}
\]

where \( i = 1, 2, 3, \ldots, L \), and the \( \in \) symbol means “a member of” in the set theory.

Although, Equation 3.1 reflects more information about DNA instance, it is unable to statistically recognize an enormous sequence. Let us consider a DNA sequence of 100 nucleotides, consequently, the possible combinations would be

\[
4^{100} = 10^\left(100 \log_4 4\right) \approx 1.6065 \times 10^{60}
\]

In case, the length of a DNA sequence is greater than 100 nucleotides then a number of different combinations will be greater than \( 1.6065 \times 10^{60} \). It is observed that the number of different combinations increased extraordinarily with the increase in the length of a sequence. Therefore, it is not feasible to establish a rational dataset for such an astronomical number of different combinations out of a sequence, which covers all the possible sequence order information statistically. In addition, the variable length of DNA sequences poses another hurdle because the operation engines can only execute vector form of equal length rather than sequential instances [36, 132].

In this regard, an attempt was made to execute sequential samples using BLAST [133] approach, however, it remains inadequate because of lack of similarity amongst the instances [134]. In order to avoid loss of sequential correlation information and encourage utilization of equal length of vector space, the concept of vector or discrete model was introduced.

Initially, a simple discrete model, Nucleic acid composition (NAC) is applied in which the frequency of each nucleotide is computed. NAC can be mathematically expressed as given in Equation 3.2.

\[
S = [f(A), f(C), f(G), f(T)]^T
\]

where \( f(A), f(T), f(C), \) and \( f(G) \) are the normalized fractions of adenine, thymine, cytosine and guanine, in DNA sequence, respectively; whereas the symbol \( T \) represents the transpose operator. However, traditional NAC does not preserve, information about sequence-order of nucleotides. Consequently, the correlation factors are neglected. To avoid loss of sequence information and incorporate the correlation factors of sequence,
the concept of pseudo amino acid composition (PseAAC) was employed, which has been utilized in almost all the areas of computational genomics and proteomics [135-139]. Accordingly, the idea of PseAAC has been extended to handle DNA/RNA sequences in the form of PseKNC [10, 140-147].

In this study, various feature extraction techniques including dinucleotide composition, trinucleotide composition and split trinucleotide composition are applied to elicit salient, prominent and variant numerical descriptors from DNA sequences.

3.2.1.1. Dinucleotide Composition

DNA primary sequence only shows information about the most adjacent local sequence-order. It can not reflect the global sequence-order information. In order to capture the global sequence-order information, dinucleotide composition (DNC) is utilized. DNC is a feature extraction technique in which the DNA sequence is expressed with the help of nucleotides pair. It computes the frequency of each combination of nucleotides pair such as the 1st dinucleotides pair is N_1N_2, the 2nd dinucleotides pair is N_2N_3 and so forth; the last dinucleotide fair is N_{L-1}N_L. Consequently, 4x4=16D corresponding features vector is generated [36, 148]. The whole process is illustrated in Figure 3.2. DNC is the simplist form of pseNAC and mathematically expressed as given in Equation 3.5.

$$S = \left[ f(AA) f(AC) f(AG) f(AT) \ldots f(TT) \right]^T$$

$$= \left[ f_1^{AA} f_2^{AC} f_3^{AG} f_4^{AT} \ldots f_{16}^{TT} \right]^T \quad (3.5)$$

where the symbol T denotes the transopose operator, \( f_1^{AA} = f(AA) \), is the normalized fraction of AA in the DNA sequence; \( f_2^{AC} = f(AC) \) is the normalized fraction of AC; \( f_4^{AT} = f(AT) \) is the normalized fraction AT in the DNA sequences and so forth. The step wise operation of DNC described in the following algorithm:

Algorithm DNC

Step-1 Input: Enter DNA sequences.
Step-2 Calculate the frequency of nucleotide pair using Equation 3.5.

Step-3 Output: $4 \times 4 = 16$D corresponding features vector is produced.

3.2.1.2. Trinucleotide Composition

Trinucleotide composition (TNC) is a feature extraction technique, in which the DNA sequence is formulated with the help of a triplet consisting of three nucleotides. In this technique, the frequency of triplets is computed. For instance, in DNA sequence, $N_1N_2N_3$, is the first triplet of trinucleotide, $N_2N_3N_4$, is the second triplet of trinucleotide, and $N_{L-2}N_{L-1}N_L$ is the last triplet of trinucleotide [36, 141]. TNC process is shown in Figure 3.3 i.e., accordingly, $4 \times 4 \times 4 = 64$D corresponding features vector is originated. The TNC mathematically expressed as given in Equation 3.6.

$$S = \left[ f (AAA) , f (AAT) , f (AAC) , f (AAG) , ..., f (TTT) \right]^T$$

$$= \left[ f_1^{\text{tri}}, f_2^{\text{tri}}, f_3^{\text{tri}}, f_4^{\text{tri}}, ..., f_{64}^{\text{tri}} \right]^T$$

(3.6)
where symbol $T$ represents the transpose operator, $f_{1}^{tri} = f(AAA)$ is the normalized fraction of AAA, $f_{2}^{tri} = f(AAC)$ is the normalized fraction of AAC, and $f_{4}^{tri} = f(AAG)$ is the normalized fraction of AAG in DNA sequences; and so forth. The step wise operation of TNC described in the following algorithm:

**Algorithm TNC**

Step-1 Input: Enter DNA sequences.

Step-2 Calculate the frequency of triplet nucleotide using Equation 3.6.

Step-3 Output: $4 \times 4 \times 4 = 64$D corresponding features vector is produced.

![Figure 3.3 Shows the process of TNC.](image)

### 3.2.1.3. Split Trinucleotide Composition

Proteins have vital informative peptides on C- and N-termini. So, direct identification of these informative peptides is not an easy job. In case of whole sequence composition, sometimes nonessential amino acids or nucleotides are dominant on the other amino acids or nucleotides. Subsequently, the targeted patterns or motifs remained unidentified, which may cause misclassification. Split amino acid composition (SAAC)
is the way through which sequence is decomposed into various segments, and composition of each segment is carried out in order to exploit the hidden information [149-152]. It decomposes the sequence at N- and C- termini. In SAAC, decomposition of sequence depends on the length of sequence. To extend the concept of SAAC from protein to DNA, split trinucleotide composition (STNC) is used.

STNC is a feature extraction technique in which the biological or DNA sequence is split into more than one segment where composition of each segment is performed independently. In this work, DNA sequence is decomposed into three distinct parts: 25 residues from start, 25 residues from last and region between these two portions. Then nucleotide composition of each part is computed separately. Consequently, 64+64+64=192D corresponding feature space is obtained that is formulated in Equation 3.7.

$$D = \left[ f_1^N, f_2^N, \ldots, f_{64}^N, f_1^{\text{int}}, f_2^{\text{int}}, \ldots, f_{64}^{\text{int}}, f_1^C, f_2^C, \ldots, f_{64}^C \right]^T \quad (3.7)$$

where N-terminus represents the first portion, C-terminus denotes the last portion and int-terminus signifies the intermediate region. These components are mathematically formulated as given in Equations 3.8, 3.9, and 3.10.

$$f_1^N, f_2^N, \ldots, f_{64}^N = f(\text{AAA}), f(\text{AAC}), f(\text{AAG}), \ldots, f(\text{TTT}) \quad (3.8)$$

$$f_1^{\text{int}}, f_2^{\text{int}}, \ldots, f_{64}^{\text{int}} = f(\text{AAA}), f(\text{AAC}), f(\text{AAG}), \ldots, f(\text{TTT}) \quad (3.9)$$

$$f_1^C, f_2^C, \ldots, f_{64}^C = f(\text{AAA}), f(\text{AAC}), f(\text{AAG}), \ldots, f(\text{TTT}) \quad (3.10)$$

Here, $$f(\text{AAA}), f(\text{AAC}), \ldots, f(\text{TTT})$$ are the normalized fractions of AAA, AAC up to TTT, respectively, in the C-terminus, int-terminus and N-terminus of the DNA sequence.

### 3.2.2. Feature Extraction for Transmembrane-Helix

This section consists of those feature extraction techniques that are applied for transmembrane helix.
Position Specific Scoring Matrix

Position specific scoring matrix (PSSM) is an evolutionary profiles method commonly used for identification of patterns or motifs in biological sequences. It gives various alignments information regarding protein families. PSI-BLAST [153-155] is commonly applied for computing PSSM profiles in order to recognize remote information about homologous proteins. After executing PSI-BLAST, 20 numerical scores are generated against each residue of protein sequence that indicates the fractions of substitution detected at a particular position in a protein family. PSSM matrix contains both positive and negative values where a positive value indicates that the substitutions take place more frequently and negative values show that amino acids are mutated less frequently in the alignment. The PSSM for a protein sequence $P$ with $L$ residues can be expressed as follows given in Equation 3.11.

$$P_{PSSM} = \begin{bmatrix}
Z_{1 \rightarrow 1} & Z_{1 \rightarrow 2} & \ldots & Z_{1 \rightarrow j} & \ldots & Z_{1 \rightarrow 20} \\
Z_{2 \rightarrow 1} & Z_{2 \rightarrow 2} & \ldots & Z_{2 \rightarrow j} & \ldots & Z_{2 \rightarrow 20} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
Z_{i \rightarrow 1} & Z_{i \rightarrow 2} & \ldots & Z_{i \rightarrow j} & \ldots & Z_{i \rightarrow 20} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
Z_{L \rightarrow 1} & Z_{L \rightarrow 2} & \ldots & Z_{L \rightarrow j} & \ldots & Z_{L \rightarrow 20}
\end{bmatrix} \quad (3.11)$$

Where $Z_{i \rightarrow j}$ shows the substitution scores of the residue $i$ replaced by $j$ during a biological evolutionary process. Here values of $j=1\ldots 20$ shows the order of amino acids.

The $P_{PSSM}$ is derived from PSI-BLAST [156, 157], exploring the Swiss-Prot database in total number of three repetitions with the cutoff $E$-value 0.001 for multiple sequences alignment. Finally, $L \times 20$ scoring matrix is produced that is further normalized using standard conversion method using Equation 3.12

$$R_{i\rightarrow j} = \frac{R_{i\rightarrow j} - \bar{R}_{i}}{SD\left(\bar{R}_{i}\right)} \quad (i=1,2, ..., L; j=1,3, ..., 20) \quad (3.12)$$
where $R^0_{ij}$ shows the actual scoring matrix produced by PSI-BLAST [22, 74, 158]. The $\bar{R}^0$ shows the mean of $R^0_{ij}$ and $SD\left(R^0\right)$ is the standard deviation over ($j=1, 2, 3, ..., 20$). The dimension of $P_{PSSM}$ for each protein sequence is variable because $L$ length matrix is generated corresponding $L$ length protein sequence. So, the development of a predictor with the ability to handle variable length of protein sequences is a complicated task. In order to bear the PSSM matrix in equal length, the concept of simple amino acid composition was applied. A protein query $P$ can be shown using Equation 3.13.

$$\hat{P}_{EVO} = \left[ \hat{R}_1 \ \hat{R}_2 \ \cdots \ \hat{R}_{20} \right]^T$$

(3.13)

$$\hat{R}_j = \frac{1}{L} \sum_{i=1}^{L} R_{ij} \ \ (j=1, 2, 3, ..., 20)$$

(3.14)

However, the major issue reported here is a loss of all the sequential ordering information in the biological evolutionary process. Further, the idea of PseAA composition was applied in order to preserve the correlation factors [159-162]. The protein sequence $P$ can be represented as given in Equations 3.15 and 3.16.

$$P^\lambda_{PseEvo} = \left[ R_1 \ \hat{R}_1 \ \cdots \ \hat{R}_{20} \ \hat{R}_2 \ \cdots \ \hat{R}_{20} \right]^T$$

(3.15)

$$R^\lambda_j = \frac{1}{L-\lambda} \sum_{i=1}^{L-\lambda} \left[ R_{i\rightarrow j} - \bar{R}_{(i+\lambda)\rightarrow j} \right]^2 \ (1, 2, 3, ..., 20; \ \lambda < L)$$

(3.16)

where

- $R^1_j$ shows the value of first-rank correlation factor by calculating the PSSM values for the adjacent neighbors.

- $R^2_j$ shows the second-rank correlation factor calculating the frequency of the second closest neighboring PSSM scores; and so on.

- The lambda $\lambda$ value should be smaller than the shortest length of sequence in the benchmark dataset.
The step wise operation of PSSM described in the following algorithm:

**Algorithm PSSM**

2. Select protein databases i.e., UniProt Knowledgebase.
3. Input: Enter protein sequence in FASTA format.
4. Set parameter (PSSM E-Vaule Cut-off) by default is $1.0e^{-3}$
5. Submit job.
6. Set the threshold value and run next iteration.
8. Output: $L \times 20$ corresponding features matrix is generated.

### 6-letter Exchange Group Representation

Proteins are constituted of distinct twenty amino acids, which are connected to each other in a linear fashion like beads in a chain. Amino acids are generally expressed by single letter code namely: A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, and Y. The structure of some amino acids is similar because of its nature. On the basis of these similarities, amino acids are distributed into six distinct groups known as 6-letter exchange group representation. In this technique, initially, the amino acids are substituted by the corresponding group, which is shown in Table 3.1 and was obtained using PAM matrix [132, 163]. For instance, all R, H, and K amino acids are substituted in the protein sequence by $a_1$, E, D, Q, and N are substituted by $a_2$, and C is replaced by $a_3$ and so forth. After replacing, each residue of an original protein sequence by the corresponding 6-letters including $a_1$, $a_2$, $a_3$, $a_4$, $a_5$, and $a_6$, the resultant sequence will be the polymer of these 6 different letters depicted in Equation 3.17. Then, the method of sliding window is employed accordingly, 6 features are generated against each location. Further, the sliding window is shifted to the next position in the protein sequence; this process is executed till the last position of the sequence.

\[
p_i = (a_1, a_2, a_3, a_4, a_5, a_6) \quad (3.17)
\]

\[
S_i = \begin{bmatrix} C_0 \end{bmatrix}_{i:6} \quad (3.18)
\]
In the Equation 3.18, $C_{ij}$ represents the relative frequency of exchange group $a_j$; $j=1,2,\ldots,6$ in window $i$. The mathematical representation of the resultant matrix is given in Equation 3.19.

$$P = \begin{pmatrix} S_1^T & S_2^T & \cdots & S_{L-w+1}^T \end{pmatrix}_{6 \times L-w+1}$$  \hspace{1cm} (3.19)

Where symbol $T$ denotes transpose operator, $w$ signifies the window size and $L$ represents the length of the protein sequence.

<table>
<thead>
<tr>
<th>Group</th>
<th>Sub-group</th>
<th>Amino acids</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exchange group</td>
<td>a_1</td>
<td>H, K, R</td>
</tr>
<tr>
<td></td>
<td>a_2</td>
<td>D, N, Q, E</td>
</tr>
<tr>
<td></td>
<td>a_3</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>a_4</td>
<td>A, G, S, P, T</td>
</tr>
<tr>
<td></td>
<td>a_5</td>
<td>I, L, V, M</td>
</tr>
<tr>
<td></td>
<td>a_6</td>
<td>F, W, Y</td>
</tr>
</tbody>
</table>

### 3.3 Feature Selection Technique

After extracting numerical attributes from protein sequences. The next step is the feature selection technique in which a subset of relevant, genuine, and important features is selected. Further, the selected subset of the feature space is used to check the discriminative power of the system. The ultimate aim of the feature selection technique is to capture more relevant information, reduce noisy data, remove unnecessary and redundant features, and decrease the computational cost with respect to space and time. In addition, it also reduces training time and enhances generalization of computational model by controlling over fitting particle swarm optimization, which is discussed as follows.

The particle swarm optimization (PSO) was given by Kennedy and Eberhart in 1995 [163], which was inspired by the social acts found in various species. The basic idea was to exploit simple analogs of social communications in order to obtain computational intelligence and never to analyze purely individual cognitive
capabilities. Many variations of PSO have been developed to improve speed to convergence, accuracy, and balance between exploitation and exploration. PSO is optimization algorithm, which is widely used for nonlinear function optimization, fuzzy system control, pattern recognition and artificial neural network training [163, 164].

In PSO, the particles are mobilized randomly in the search space in order to achieve an objective function for minimization. The objective function for each particle computed for its current position. The displacement of each particle is obtained using the search space by storing the log of its current position as well as the best positions among those elements within the available swarm with few random perturbations. The later iteration occurs as soon as the movement of all particles in the given swarm is done. The step wise operation of PSO described in the following algorithm:

**Algorithm PSO**

Steps:
1: Initialization: Particles are assigned random positions and velocities.

2: Repeat 3-7

3: For each individual particle, calculates the fitness value or function.

4: Analyze particle’s current fitness value to the pBest. Check if the existing value is greater than pBest, then assigns current fitness to pBest otherwise keep previous pBest.

5: Assign value to gBest by finding amongst the neighborhood for the best particle.

6: The position and velocity for each particle are computed as:

\[
\begin{align*}
    v_i &\leftarrow v_i + U(0,\phi_1) \oslash (p_i - x_i) + U(0,\phi_2) \oslash (p_g - x_i) \\
x_i &\leftarrow x_i + v_i
\end{align*}
\]

(3.20)

7: Exit the loop if the criterion is met; otherwise go to step-3.

8: End
In the above Equation 3.20 and algorithm, the $x_i$ represents the current location in a search space using set of coordinates defining a point, $p_i$ represents the previous best location or position, $v_i$ denotes the velocity, $U(0,\varnothing_i)$ signifies random numbers for a vector evenly placed across in $[0,\varnothing_i]$ randomly obtained at each repetition for the particles and $\otimes$ represents a component-wise multiplication. The resultant score of the best function so far is assigned to a variable known as pBest (previous best), which is used for assessment on later iterations. The main purpose is to find consistently the appropriate locations and update $p_i$ and pBest, respectively. Using addition of $v_i$ coordinates to $x_i$ for the selection of new positions for a particular point, the algorithm manipulates it by fixing $v_i$ successfully by considering as step size. Moreover, the velocities $v_i$ of particles are confined within the range of $[-V_{\text{min}}, +V_{\text{max}}]$.

### 3.4 Data Validation (Partition)

Before going to train the model it is essential to partition the data into various folds. For this purpose, generally two well established statistical techniques are used namely; holdout and cross validation. In holdout, data technique particulars usage of data for testing while the remaining data is utilized for training. Generally, about one third part of the data is used for conduction of test and remaining two third part of the data are reserved for training. However, the key issue reported in holdout technique is that it has no surety to properly represent all the instances of the given data. It might be expected that in training phase few of the classes will have no representation, could presume that a hypothesis of missing data may be learned.

The second data partitioned technique is a cross-validation. This is mostly utilized for evaluating the hypotheses performance. Using this technique, the whole dataset is partitioned into a number of fixed mutually exclusive folds. Usually, cross-validation test is distributed into four types namely; jackknife, independent, self-consistency and subsampling dataset tests.
3.4.1. Jackknife Test

Jackknife cross validation is extensively applied test in the domain of machine learning, and pattern recognition related applications. In this method, the overall dataset is divided or fragmented into $n$-fold, also known as leave-one-out cross validation test. Moreover, only one instance is reserved for testing while the remaining $n-1$ instances of dataset are utilized for training, the overall process is executed $n$ times. It produces a unique outcome, due to which it is considered to be one of the efficient validation tests among these. Jackknife test remains dominant over other cross validation tests for the main two reasons. The first one is, maximum amount of data is used for training the model and as a result, the generalization power of hypothesis is increased. The second is, it has no random sampling, where the result produced by this test always remains unique for a particular dataset. Hence, due to such effective characteristics, Leave-one-out test is broadly utilized by researchers to investigate the classification rate of learning hypothesis. The main limitation of jackknife test is its computational cost because it executes for $n$ times.

3.4.2. Sub-sampling Test

In this test, $k$-fold is created by splitting the data. In this type of validation test, the single fold is reserved for testing, while the remaining $k-1$ folds are utilized for training of a model. This process is executed $k$-times, where for testing every fold is used at least once. Five-fold, seven-fold and ten-fold sub-sampling tests are usually used, but ten-fold is commonly applied. The main drawback of sub-sampling test is the selection of small portion of data, where some classes remain without consideration during training phase. Furthermore, numerous selections will always produce various outcomes even for the same predictor and same dataset. Therefore, the sub-sampling test doesn’t produce a distinctive result for a particular dataset.

3.5 Learning Hypotheses

Classification is the sub-class of artificial intelligence and machine learning, which classifies the novel data into predefined classes [165]. Classification is performed by two ways: supervised and unsupervised learning. In supervised learning, the target labels are already defined whereas in unsupervised learning, the predefined labels are
not provided. Classification process can be accomplished in two phases i.e., training phase and testing phase. In the first phase, the learning hypothesis memorizes about the objects from the attributes or pattern obtained during feature extraction. In the later phase, the novel instances are classified on the basis of learning or observations. Various intelligent learning approaches have been carried out to classify the data in the domain of data mining, pattern recognition, computer vision, and bioinformatics. In this research study, several leading and distinguished supervised learning hypotheses are applied.

3.5.1. Support Vector Machine

SVM is a supervised learning algorithm widely applicable in the domain of classification and regression. It was first proposed by Cortes and Vapnik in 1995 for binary problems, but view to the importance of multi-class problems it was latterly updated by Vapnik in 1998 [10, 166-169]. SVM transforms the provided vector space to high dimensional vector space and then draws a separating line between two classes’ instances. Further, parallel lines are drawn to the separating line, which yields the distance between closest points and separating line in the training dataset; the points are known as support vectors and the distance represents the margin illustrated in Figure 3.4. SVM utilizes various kernels such as radial base function, polynomial, linear and sigmoid to maximize the space among instances of two classes in order to correctly classify [170]. The main goal of SVM is to search optimize separating hyperplane, which maximizes the distance from the separating line to the points closest to it on both sides.

\[
\min \Psi(\omega, \xi) = \frac{1}{2} \| \omega \|^2 + C \sum_{i=1}^{n} \xi_i \\
\text{Subject to} \quad y_i[(\omega, x_i) + b] - 1 + \xi_i >= 0
\]  

(3.21)

where:

- \((x_i, y_i), i=1, 2, 3, \ldots, n, x_i \in R^d, y_i \in (1, -1)\) are the training samples,
• C showing capacity constant controlling the trade-off between minimizing the errors and maximizing the margin

• \( \xi \) shows parameters for handling non-separable input data.

The decision function is

\[
f(x) = \text{sign}((\omega \cdot x) + b) = \text{sign}\left(\sum_{i=1}^{n} (a_i^* y_i (x_i, x) + b^*)\right)
\] (3.22)

The said optimization problem is also solved by the dual problem as follows:

\[
\text{max } Q(a) = \sum_{i=1}^{n} a_i = \frac{1}{2} \sum_{i,j=1}^{n} a_i a_j y_i y_j (x_i, x_j)
\] (3.23)

Subject to

\[
\sum_{i=1}^{n} a_i y_i = 0 \text{ and } 0 \leq a_i \leq C, \quad i = 1, 2, 3, ..., n
\]

The \((x_i, x_j)\) represents the kernel, where coefficients \(a_i^*\) must be non-zero in the decision function.

According to the concept of kernels, researchers have suggested various types of kernels to calculate the inner product efficiently. \(C\) and \(\gamma\) as input parameters are taken by SVM with RBF/ sigmoid kernel. In polynomial kernel \(d\) is an additional parameter, which shows the polynomial degree. Moreover, SVM along with linear kernel uses only \(C\) parameter. Linear kernel can be expressed as given by:

\[
K(x, y) = \exp(-\gamma \|x, y\|^2)
\] (3.24)

In linear kernel, the computational process is faster because SVM never converts the original space into a high dimensional vector space. Polynomial kernel can be expressed as given in Equation 3.25

\[
K(x, y) = (x \cdot y + 1)^d
\] (3.25)

In polynomial kernel the hyperplane is depending on \(d\), which control the complexity in the input vector space. Where \(d\) denotes the degree of polynomial kernel. If \(d=1\), the
polynomial kernel works like linear kernel. The RBF kernel is numerically formulated as shown below:

\[ K(x, y) = \exp(-\gamma \|x - y\|^2) \]  

(3.26)

In Equation 3.26, \( \gamma \) is utilized to show the width of the Gaussian function. In this research study, LIBSVM library is used for conducting the experiments.

In Figure 3.4, the circles and triangle represent the training samples. The red line denotes the optimal hyperplane. Samples on the black lines are support vectors.

3.5.2. k-Nearest Neighbor

The concept of \( k \)-nearest-neighbor (KNN) was broadly adopted in the area of machine learning, data mining, and classification due to incredible performance, adaptability, simplicity, and easy to understand. It has no prior information about the distribution of the data, that why it known as non-parametric algorithm \([171]\). KNN has no explicit training data so it tends to keep all the training data in the testing phase. It classifies the novel instance on the basis of nearest neighbors by using Euclidean distance. KNN is
also called as instance base learner or lazy learner. It makes decision on the basis of calculating the distance amongst a query scenario as well as a set of scenarios in the training dataset [172, 173]. The Euclidean distance is calculated for the two tuples or points as:

\[
E_{dis}(a, b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}
\]  

(3.27)

Suppose, we have two classes, i.e. class A and class B which represented by the blue color triangle and the brown color circle where the red star is a novel shape not categorized shown in Figure 3.5. In order to categorize the novel shape, Euclidean distance is computed for k=3 neighbors. Among these three closest neighbors, one belongs to Class A and two belong to Class B, hence, on the basis of majority voting B class is assigned to red star shape.

![Figure 3.5 Example of K-Nearest Neighbor Algorithm.](image)

### 3.5.3. Probabilistic Neural Network

D.F. Specht incorporated the idea of probability with neural network by introducing probabilistic neural network (PNN) [174-176]. It is widely utilized for classification in machine learning related applications. The most effective and interesting property of
PNN is that, it has the ability to represent any number of inputs/output complex relationships. The simplicity and transparency of PNN are also similar to conventional statistical classification approaches [169, 177]. It operates in completely parallel way, due to which it does not need feedbacks for the input from the individual neurons. Figure 3.6 displays the architecture of PNN containing four layers namely: input layer, pattern layer, summation layer and output layer.

The first layer consists of distant $N$ nodes, these nodes or states are entirely interleaved to $M$ nodes of the PNN in the second layer. In the third layer, a particular node relates to only a training object. $P_i$ is used as an input vector by pattern node $j$ by using an activation function.

$$u_{ij} = \exp \left(-\frac{1}{2} \frac{\|P_i - P_j\|^2}{\sigma^2} \right)$$  \hspace{1cm} (3.28)

where $u_{ij}$ represents the outcome for the pattern node $j$ and $\delta$ is a smoothing factor that limits the width of the activation function.

As the distance increases of $|P_j - P|$ amongst the inputted vectors $P_i$ and $P_j$ of the pattern node $j$, decrease is observed in resemblance between the two data vectors conversely. The pattern layer yields are used as input for the third layer of PNN i.e., summation layer, which has $v$ competitive nodes each pointing to one class. Hence making each summation node $v$ connecting to the pattern nodes that are related to the training objects of class $v$. 
The input for the output layer is computed using summation layer by receiving outputs from associated pattern nodes:

$$ f_v(P_i) = \frac{1}{N_v} \sum u_i $$  

(3.29)

In Equation 3.29, $Q_v$ represents the label of the class relating to the summation node $v$ whereas $N_v$ represents the number of training instances of the same class. For example, if all data vectors are normalized; the Equation 3.29 can be mathematically expressed as:

$$ f_v(P_i) = \frac{1}{N_v} \sum \exp \left( \frac{(P_i P^r_i - 1)}{\sigma^2} \right) $$  

(3.30)

The outputs of the summation layer can be calculated as the posterior class membership probabilities:

$$ P(Q_v = v | P) = f_v(P) / \sum_{i=1}^{V} f_i(P) $$  

(3.31)
Using Equation 3.31, output layer is added with a classification rule for assigning the input vector to a particular class $P_i$. The direct method is used to select a class whose $P(v|P_i)$ is maximum.

### 3.5.4. Fuzzy Support Vector Machine

SVM is a widely used tool for solving classification problems, but there are still some limitations such as unclassifiable regions and undefined multi-label regions. Resolving these limitations, the concept of fuzzy was incorporated with support vector machine [178, 179]. Traditional SVM transforms $n$-classes into $n$-binary classes in case of multiclass problem, which considers the $i^{th}$ class as separate class from the remaining classes. Suppose, the separating line, which categorizes the $i^{th}$ class from remaining classes, can be represented as:

$$ D_i(Z) = w_i^T + b_i $$

(3.32)

where:

- If $D_i(Z) = 0$ shows the best possible hyperplane,

- If $D_i(Z) = 1$ is satisfied the instances relating to the class $i$

- If $D_i(Z) = -1$ is satisfied by the instances relating to the remaining classes.

- If $D_i(Z) > 0$ is true for one $i$ class then categorization of $z$ is mapped into class $i$.

- In contrast, if $D_i(Z) > 0$ is satisfied for no $i$ or more $i$’s classes then $z$ is unclassifiable.

To solve these limitations fuzzy member function is used. Using the technique, uni-dimensional membership functions $m_i(Z)$ on the directions orthogonal to the best possible separating hyperplane $D_i(Z) = 0$ as shown below:
1. For \( i = j \)
   \[
   m_{ij}(Z) = \begin{cases} 
   1 & \text{for } D_i(Z) > 1 \\
   D_i(Z) & \text{otherwise}
   \end{cases}
   \]
   \[(3.33)\]

2. For \( i \neq j \)
   \[
   m_{ij}(Z) = \begin{cases} 
   1 & \text{for } D_i(Z) < -1 \\
   -D_i(Z) & \text{otherwise}
   \end{cases}
   \]
   \[(3.34)\]

In above Equation 3.33 and Equation 3.34

- \( D_i(Z) > 1 \) indicates the availability of only one class training data, so the degree of class \( i \) is equal to 1 and otherwise \( D_i(Z) \).

- For \( i \neq j \), \( D_j(Z) = 0 \) contains class \( i \) on the negative side. In this scenario, support vectors don’t contain the data of class \( i \) but as \( D_i(Z) < -1 \); it is assumed that the degree of membership of class \( i \) is 1 and otherwise \( -D_j(Z) \).

We define the membership function of class \( i \) if \( Z \) having minimum operator for \( m_{ij}(Z) \), \( j = 1, 2, 3, \ldots, n \)

\[
m_i(Z) = \min_{j=1,2,3,\ldots,n} \left[ m_{ij}(Z) \right]
\]
\[(3.35)\]

Currently, the datum \( Z \) is categorized into the class

\[
\arg \max_{i=1 \ldots n} m_i(Z)
\]
\[(3.36)\]

If \( x \) satisfies

\[
D_k(Z) = \begin{cases} 
> 0 & \text{for } i=k \\
\leq 0 & \text{for } i \neq k \text{ where } k = 1, \ldots, n
\end{cases}
\]
\[(3.37)\]

From Equations 3.33 and Equation 3.34 \( m_i(Z) > 0 \) and \( m_j(Z) \leq 0 \), if \( j \neq i \) where \( j = 1, 2, 3, \ldots, n \) hold, \( x \) is categorized into class \( i \).

Now suppose \( D_i(Z) > 0 \) is satisfying \( i_1, i_2, \ldots, i_L > 1 \) then using Equations 35-37, \( m_k(Z) \) is specified as follow
\[ k \in \{i_1, \ldots, i_L\} \]

1. \[ m_k(Z) = \min_{(j=i_1, \ldots, i_L)} -D_j(Z) \] \hspace{1cm} (3.38)

2. \[ m_i(Z) = \min_{(j=i_1, \ldots, i_L)} -D_j(Z) \] \hspace{1cm} (3.39)

Hence achieving the maximum degree of membership among \( m_k(Z), k = i_1, \ldots, i_L \) \( D_k(Z) \). Namely, \( D_k(Z) \) is maximized in \( k \in \{i_1, \ldots, i_L\} \). Suppose \( D_i(Z) > 0 \) is not satisfied for any class then:

\[ D_i(Z) < 0 \quad \text{for } i=1,2,3,\ldots,n \] \hspace{1cm} (3.40)

Then Equation 3.34 is given

\[ m_i(Z) = D_i(Z) \] \hspace{1cm} (3.41)

The process of classification is given below

1. For \( X \), if \( (D_i(Z) > 0) \) is fulfilled only for single class then the input is classified into the class. Else go to step 2.

2. If \( (D_i(Z) > 0) \) is fulfilled for multi class \( i \) \( (i=i_1, \ldots, i_L, L>1) \) categorizes class using the datum into the maximum \( D_i(Z)(i \in \{i_1, \ldots, i_L\}) \) else go to step 3.

3. If \( (D_i(Z) \leq 0) \) is fulfilled for whole the classes, categorizes class using the datum into the minimum absolute value of \( D_i(Z) \).

### 3.6 Evaluation criteria

The performance of hypotheses is assessed in term of classification rates. Classification rate demonstrates the strength or weakness of hypothesis. Various measures are used for showing the classification rates. These measures are computed from confusion matrix, which contains the predicted results and actual results. In confusion matrix, each row defines the actual label while each column determines the predicted label for that class.
Table 3.2 A Confusion Matrix

<table>
<thead>
<tr>
<th>Predicted Label</th>
<th>Positives</th>
<th>Negatives</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target Label:</strong></td>
<td>Positives</td>
<td>TP</td>
</tr>
<tr>
<td></td>
<td>Negatives</td>
<td>FP</td>
</tr>
</tbody>
</table>

Table 3.2 illustrate the confusion matrix where TP represents positive instances, which are predicted positive whereas FN indicates number of positive instances predicted as negatives. Likewise, FP shows negative instances predicted as positive, it is also known as type-I error whereas TN indicates negative instances predicted as negative, it is also known as type-II error. The performance assessment parameters utilized in this research work is discussed below:

### 3.6.1. Accuracy

To examine the performance of computational models, accuracy is the foremost parameter to evaluate it. It measures the true prediction of a model in term of TP and TN. It is the ratio of the sum of TP and TN instances to the whole instances in percentage. Accuracy can be computed using Equation 3.42

\[
Accuracy = \frac{TP + TN}{TN + FN + TP + FP} \times 100 \quad (3.42)
\]

### 3.6.2. Sensitivity

Sensitivity is a performance parameter which shows the proportion of true positive; therefore, it considered the recall rate or true positive rate. Sensitivity shows the ratio between the predicted TP instances and whole number of TP instances.

\[
Sensitivity = \frac{TP}{TP + FN} \times 100 \quad (3.43)
\]

### 3.6.3. Specificity

Specificity is the proportion of true negative, it is also called true negative proportion. Specificity gives the ratio between the predicted TN instances and total number of TN instances, respectively.
3.6.4. Mathew’s Correlation Coefficient

Matthew’s correlation coefficient (MCC) is one of the standards and reliable performance measure parameter in classification [126, 180]. It evaluates the quality of a prediction model [180]. MCC basically converts a confusion matrix into a scalar value in the range of [-1, +1], where +1 assures that the hypothesis learner produces correct prediction, 0 value shows that the hypothesis learner yields average random prediction, and -1 indicates the hypothesis generates incorrect predictions. MCC can be computed using Equation 3.45.

\[
MCC(i) = \frac{TP \times TN - FP \times FN}{\sqrt{[FP + TP][FN + TP][FN + TN][FP + TN]}}
\] (3.45)

MCC is a dominant quantitative measure specifically in the situations where imbalance benchmark dataset is available to the hypothesis learner. Hence, MCC is capable of solving the challenges faced by accuracy. For instance, if the positive instances are greater than negative instances, the hypothesis learner may predict at ease the whole examples as positive because its bias towards majority class. Therefore, the predicted outcomes of the hypothesis learner will become worse for negative class. In such scenario, 100% accuracy is noticed for positive class whereas MCC is reported 0%.

3.6.5. F-measure

The statistical measure based on harmonic mean of both the recall and precision of the test is known as F-measure. Where precision denotes true predication’s percentage among all returned predictions, and recall shows true prediction’s percentage among total number of observed instances [181-183]. F-measure generates its output in the range of 0 and 1, having output approaching close to 0 showing worst performance. Otherwise, the output approaching close to 1 indicates best performance.

\[
\text{Precision} = \frac{TP}{FP + TP}
\] (3.46)
Recall = \frac{TP}{FN + TP} \quad (3.47)

F - measure = \frac{[\text{Precision} \times \text{Recall}]}{\text{Precision} + \text{Recall}} \times 2 \quad (3.48)
Chapter 4

iNuc-STNC: A SEQUENCE-BASED PREDICTOR FOR DISCRIMINATION OF NUCLEOSOME POSITIONING IN GENOMES

4.1 Introduction

In previous chapters, a discussion was carried out regarding literature and implemented approaches to the research. This chapter presents a detailed discussion about the first phase of research. In the first phase of research, nucleosome positioning in genomes is targeted because the nucleosome is a vital repetitive unit of eukaryotic chromatin, composed of DNA wrapped around a histone core. Having thorough discussions on the drawbacks of traditional systems yielding highlighted need of contemporary computational approaches in the literature review. Due to technological improvements, various researchers endeavored to adopt the concept of advanced approaches in order to reduce the risk of traditional methods. Despite tremendous achievement has been obtained through pattern recognition and machine learning there is still the room for improvement in terms of performance still exist. In view of this, an effective, vigorous and high throughput sequence based intelligent computational model iNuc-STNC is developed for the discrimination of nucleosome positioning in genomes.

The experimental results reveal that our proposed model achieved show encouraging results and predicted nucleosome positions in genomes with high confidence. In the proposed iNuc-STNC model, DNA sequences were expressed using various discrete feature extraction techniques namely: DNC, TNC and STNC. Several learning hypotheses were used to select the best one for prediction. The discrimination power of these learning hypotheses was evaluated by applying statistical cross validation test i.e., jackknife test. The success rate of the proposed predictor iNuc-STNC is evaluated in terms of sensitivity, accuracy, MCC and specificity respectively because these are
important measures for assessing any supervised learning methods. Figure 4.1 represents the framework of iNuc-STNC model.

![Figure 4.1 Framework of iNuc-STNC Model](image)

4.2 Materials and Methods

In this section, the dataset will be explained, then the feature extraction techniques will be elucidated and finally, results of the proposed model will be discussed in detail.

4.2.1. Datasets

This study contained three various species, i.e., C.elegans, H.sapiens and D.melanogaster. The dataset about these species was downloaded from various sites as mentioned in Table 4.1. The H.sapiens genomes along with its nucleosome map have a
significant volume of data. As per Liu et al., nucleosome-forming and nucleosome-inhibiting sequences were extracted from chromosome 20 [184]. Nucleosome-forming is a positive class whereas, the nucleosome-inhibiting is a negative class. For remaining two species such as, D. melanogaster and C. elegans, the nucleosome-forming and nucleosome-inhabiting data were obtained or extracted from their whole genomes. Each of the DNA segments was allocated a nucleosome formation value to reflect its propensity in forming a nucleosome, i.e., greater the value, more likely the segment will form a nucleosome.

According to Chou’s, a dataset with several repetitive samples having a high resemblance would a shortage of statistical representativeness. If a predictor is trained and tested with such a biased dataset, consequently, may produce misclassification rate with a low precision [150, 185]. To eliminate these discrepancies, the CD-HIT software [186], with a cutoff threshold of 80% is applied, in order to minimize homologous between DNA sequence. As result three benchmark datasets are obtained which are expressed below

\[ S_1 = S_1^+ U S_1^- \]  \hspace{1cm} (4.1)

\[ S_2 = S_2^+ U S_2^- \]  \hspace{1cm} (4.2)

\[ S_3 = S_3^+ U S_3^- \]  \hspace{1cm} (4.3)

The benchmark dataset \( S_1 \) for H. sapiens contains 4573 sequences in which positive dataset \( S_1^+ \) has 2273 nucleosome-forming sequences whereas, the negative dataset \( S_1^- \) has 2300 nucleosome-inhabiting sequences. The second benchmark dataset \( S_2 \) for C. elegans contains 5175 sequences in which \( S_2^+ \) has 2567 nucleosome-forming sequences whereas, \( S_2^- \) has 2608 nucleosome-inhabiting sequences. Similarly, the third benchmark dataset \( S_3 \) for D. melanogaster is comprised of 5750 in which \( S_3^+ \) has 2900 nucleosome-forming sequences whereas, \( S_3^- \) has 2850 nucleosome-inhabiting sequences. The symbol \( U \) represents the union of two set [10].
4.2.2. Feature Extraction Techniques

In this work, we have converted DNA sequences into numerical descriptors by using three powerful and feature extraction schemes, i.e., DNC, TNC, and Split TNC in order to elicit salient, propound and high variated features. Detail discussions about these feature extraction schemes were presented in Chapter 3.

4.3 Results and Discussion

In this study, the classification rate of various learning hypotheses was examined by using jackknife cross validation test. Jackknife test is applied here because it is considered as an excellent and effective test due to its unique output generation. In this test, each instance of the dataset takes a turn as a testing instance. In addition, at the same time, a huge amount of data is used for training the model. Various metrics namely: sensitivity, specificity, accuracy, and MCC are applied to measure the predictive quality of each learning hypothesis.

4.3.1. Performance Comparison of learning hypotheses on various feature spaces using Dataset \( S_1 \)

Table 4.2 presented the experimental results of iNuc-STNC model for the dataset \( S_1 \) using various feature spaces, i.e., DNC, TNC and STNC, along with three different learning hypotheses. After examining the results of DNC feature space, it is observed that the accuracy of all the three learning hypotheses is comparatively similar. Among these, SVM has yielded the highest outcomes, which are 79.96% of accuracy, 85.04% of sensitivity, 74.95% of specificity, and 0.60 of MCC. Similarly, the obtained results of KNN is better in term of MCC than that of SVM and PNN. It has achieved, which are 79.57% accuracy, 90.18% sensitivity, 69.08% specificity, and 0.66 MCC. Further,
the results of the next feature space TNC is examined which demonstrated that the true
classification rates of learning hypotheses are enhanced, which exhibited that as the size
of pair increased the discrimination power of learning hypotheses improved. It reflects
more order information and correlation factors of DNA sequence. Among using
learning hypotheses, SVM obtained the highest results in term of all measures i.e., a
sensitivity of 87.68%, an accuracy of 86.51%, specificity of 85.34%, and MCC of 0.73.
On the other hand, the success rates of PNN and KNN are somehow similar, which are
84.86% and 85.52% of accuracy, 91.37% and 91.42% of sensitivity, 78.43% and
79.69% of specificity, and 0.71 of MCC. Furthermore, in order to exhibit the hidden
salient information, which was concealed due to the dominance of irrelevant
information, the sequence is split into various parts and then TNC is applied. It is known
as STNC. After exploring the results of STNC feature space, SVM has obtained
outstanding success rates compared to PNN and KNN. It has achieved a sensitivity of
89.31%, accuracy of 87.60%, specificity of 85.91%, and MCC of 0.75. In contrast,
KNN and PNN have not utilized the effectiveness of STNC. They have yielded similar
results to simple TNC. The performance comparison of various learning hypothesis and
feature spaces are presented in Figure 4.2.

<table>
<thead>
<tr>
<th>Feature space</th>
<th>Learning hypothesis</th>
<th>Accuracy (%)</th>
<th>Sensitivity (%)</th>
<th>Specificity (%)</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNC</td>
<td>SVM</td>
<td>79.96</td>
<td>85.04</td>
<td>74.95</td>
<td>0.60</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>79.57</td>
<td>90.18</td>
<td>69.08</td>
<td>0.66</td>
</tr>
<tr>
<td></td>
<td>PNN</td>
<td>79.26</td>
<td>89.79</td>
<td>68.87</td>
<td>0.59</td>
</tr>
<tr>
<td>TNC</td>
<td>SVM</td>
<td>86.51</td>
<td>87.68</td>
<td>85.34</td>
<td>0.73</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>84.86</td>
<td>91.37</td>
<td>78.43</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>PNN</td>
<td>85.52</td>
<td>91.42</td>
<td>79.69</td>
<td>0.71</td>
</tr>
<tr>
<td>STNC</td>
<td>SVM</td>
<td><strong>87.60</strong></td>
<td><strong>89.31</strong></td>
<td><strong>85.91</strong></td>
<td><strong>0.75</strong></td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>85.59</td>
<td>92.96</td>
<td>78.30</td>
<td>0.72</td>
</tr>
<tr>
<td></td>
<td>PNN</td>
<td>85.32</td>
<td>92.30</td>
<td>78.43</td>
<td>0.71</td>
</tr>
</tbody>
</table>
4.3.2. Performance Comparison of learning hypotheses on various feature spaces using Dataset S₂

To validate the effectiveness of feature spaces, they are also investigated on another dataset. Table 4.3 shows the success rates of learning hypotheses using various feature spaces. Again, in case of DNC feature space, all the three learning hypotheses obtained comparatively similar accuracy. However, the success rate of SVM is considered little better compared to PNN and KNN. It has yielded 82.62% of accuracy, 88.23% of sensitivity, 77.10% of specificity, and 0.65 of MCC. Whereas, PNN, on the other hand, obtained accuracy of 81.49%, the sensitivity of 88.78%, specificity of 74.27%, and MCC of 0.63. Similarly, for TNC feature space, SVM still achieved better outcomes with an accuracy of 85.68%, sensitivity of 88.89%, specificity of 82.51%, and MCC of 0.65. Again no major change has been observed in the performance of KNN and PNN. PNN obtained 83.94% accuracy, 90.41% sensitivity, 77.57% specificity, and 0.68 MCC. Whereas KNN has yielded 83.69% accuracy, 90.65% sensitivity, 76.84%
specificity, and 0.68 MCC. Likewise, dataset 1, remarkable improvement has also been detected by applying STNC on second benchmark dataset. It is also remained extraordinary in case $S_2$ dataset. As a result, the true classification rates of learning hypotheses are enhanced. The highest success rates have been obtained by SVM, which are 88.62% accuracy, 91.62% the sensitivity, 85.66% specificity, and 0.77 MCC. Alike to DNC and TNC, the success rate of PNN and KNN are about same. PNN has yielded an accuracy of 86.78%, sensitivity of 92.05%, specificity of 81.59%, and MCC of 0.74. On the other hand, KNN has obtained 86.45% of accuracy, 92.13% of sensitivity, 80.86% of specificity and 0.73 MCC. The performance analysis of various learning hypothesis and feature spaces of dataset $S_2$ are presented in Figure 4.3.
### Table 4.3 Performance Analysis of various Feature Spaces using S2

<table>
<thead>
<tr>
<th>feature space</th>
<th>Learning hypothesis</th>
<th>Accuracy (%)</th>
<th>Sensitivity (%)</th>
<th>Specificity (%)</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PNN</td>
<td>DNC</td>
<td>81.49</td>
<td>88.78</td>
<td>74.27</td>
<td>0.63</td>
</tr>
<tr>
<td>KNN</td>
<td></td>
<td>81.33</td>
<td>89.83</td>
<td>72.96</td>
<td>0.63</td>
</tr>
<tr>
<td>SVM</td>
<td></td>
<td>82.62</td>
<td>88.23</td>
<td>77.10</td>
<td>0.65</td>
</tr>
<tr>
<td>PNN</td>
<td>TNC</td>
<td>83.94</td>
<td>90.41</td>
<td>77.57</td>
<td>0.68</td>
</tr>
<tr>
<td>KNN</td>
<td></td>
<td>83.69</td>
<td>90.65</td>
<td>76.84</td>
<td>0.68</td>
</tr>
<tr>
<td>SVM</td>
<td></td>
<td>85.68</td>
<td>88.89</td>
<td>82.51</td>
<td>0.71</td>
</tr>
<tr>
<td>PNN</td>
<td>STNC</td>
<td>86.78</td>
<td>92.05</td>
<td>81.59</td>
<td>0.74</td>
</tr>
<tr>
<td>KNN</td>
<td></td>
<td>86.45</td>
<td>92.13</td>
<td>80.86</td>
<td>0.73</td>
</tr>
<tr>
<td>SVM</td>
<td></td>
<td><strong>88.62</strong></td>
<td><strong>91.62</strong></td>
<td><strong>85.66</strong></td>
<td><strong>0.77</strong></td>
</tr>
</tbody>
</table>

**Figure 4.3** Performance of various learning hypothesis and feature spaces on dataset S2.
4.3.3. Performance comparison of Learning hypotheses on various feature spaces using dataset $S_3$

Another benchmark dataset was carried out to exhibit the strength of feature spaces. The performance of learning hypotheses using various feature spaces are listed in Table 4.4. By exploring the results of DNC, SVM has achieved better results compared to KNN and PNN. It has obtained 77.90% of accuracy, 73.59% of sensitivity, 82.28% of specificity, and 0.56 of MCC. PNN, on the other hand, produced an accuracy of 76.14%, sensitivity of 76.21%, specificity of 76.07%, and MCC of 0.52. Whereas the accuracy, sensitivity, specificity and MCC of KNN are 74.73%, 79.21%, 70.17%, and 0.49, respectively. By examining the performance of TNC feature space, still, SVM has achieved encouraging results, with an accuracy of 80.52%, sensitivity of 77.48%, specificity of 83.61%, and MCC of 0.61. Whereas the performance of PNN is relatively well. It has achieved an accuracy of 77.86%, sensitivity of 77.83%, specificity of 77.89%, and MCC of 0.55. In contrast, the performance of KNN is low than that of SVM and PNN. It has yielded 77.60% accuracy, 81.38% sensitivity, 73.75% specificity and 0.55 MCC. Similarly to S1 and S2, the learning hypotheses have improved the true classification rates and accurately identified nucleosome positioning in genomes in case of STNC. By analyzing the performance of STNC, the success rates of SVM are better compared to PNN and KNN. It has yielded an accuracy of 81.67%, sensitivity of 79.79%, specificity of 83.61%, and MCC of 0.63. The outcomes of PNN are 79.35% accuracy, 85.82% sensitivity, 72.77% specificity, and 0.59 MCC. Finally, it has been concluded that STNC in combination with SVM has achieved quite remarkable results on all the three benchmark datasets. The computational model iNuc-STNC is developed on the basis of STNC and SVM is utilized as learning hypothesis. The performance analysis of various learning hypothesis and feature spaces of dataset $S_3$ are presented in Figure 4.4.
<table>
<thead>
<tr>
<th>Feature space</th>
<th>Learning hypothesis</th>
<th>Accuracy (%)</th>
<th>Sensitivity (%)</th>
<th>Specificity (%)</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNC</td>
<td>PNN</td>
<td>76.14</td>
<td>76.21</td>
<td>76.07</td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>74.73</td>
<td>79.21</td>
<td>70.17</td>
<td>0.49</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>77.90</td>
<td>73.59</td>
<td>82.28</td>
<td>0.56</td>
</tr>
<tr>
<td>TNC</td>
<td>PNN</td>
<td>77.86</td>
<td>77.83</td>
<td>77.89</td>
<td>0.55</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>77.60</td>
<td>81.38</td>
<td>73.75</td>
<td>0.55</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>80.52</td>
<td>77.48</td>
<td>83.61</td>
<td>0.61</td>
</tr>
<tr>
<td>STNC</td>
<td>PNN</td>
<td>79.35</td>
<td>85.82</td>
<td>72.77</td>
<td>0.59</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>78.53</td>
<td>86.62</td>
<td>70.31</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>81.67</td>
<td>79.79</td>
<td>83.61</td>
<td>0.63</td>
</tr>
</tbody>
</table>

Figure 4.4 Performance of various learning hypothesis and feature spaces on dataset $S_3$. 
4.3.4. Performance Comparison of iNuc-STNC model with other Models

In Table 4.5, success rates of proposed iNuc-STNC model have also been compared with the existing iNuc-PseKNC model on examined datasets to illustrate the strength of proposed model. The pioneer work on these datasets was performed and developed iNuc-PseKNC model for prediction of nucleosome positioning in genomes [10]. Their model achieved 87.60% accuracy for $S_1$. Further, the success rate of sensitivity, specificity, and MCC were 89.31%, 85.91% and 0.75, respectively. The predicted outcomes of iNuc-PseKNC model for $S_2$ were 88.62% accuracy, 91.62% sensitivity, 86.66% specificity and 0.77 MCC. Similarly, the outcomes for $S_3$ were 81.67% accuracy, 79.76% sensitivity, 83.61% specificity and 0.63 MCC. In contrast, our proposed model iNuc-STNC has obtained 87.60% of accuracy for $S_1$. Furthermore, the values of sensitivity, specificity and MCC are 89.31%, 85.91% and 0.75, respectively. For $S_2$, our model yielded accuracy of 88.62%, sensitivity of 91.62%, specificity of 86.66%, and MCC of 0.77, respectively. Similarly for $S_3$, accuracy is 81.67%, sensitivity is 79.76%, specificity is 83.61% and MCC is 0.63. After empirical evaluation, it is observed that iNuc-STNC model has obtained astonishing results than that of the current state of arts in the literature so far. All these achievements is credited to STNC by exploring the concealed salient pattern/motifs, which make it easy for learning hypothesis to accurately identify nucleosome positioning in genomes. It was also possible due to the generalization power of SVM because its map provided input space to high dimensional space, where the discrimination can be done very easily between classes.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Species</th>
<th>Model</th>
<th>Accuracy (%)</th>
<th>Sensitivity (%)</th>
<th>Specificity (%)</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>S₁</td>
<td>H. sapiens</td>
<td>iNuc-PseKNC</td>
<td>86.27</td>
<td>87.86</td>
<td>84.70</td>
<td>0.73</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iNuc-STNC</td>
<td><strong>87.60</strong></td>
<td><strong>89.31</strong></td>
<td><strong>85.91</strong></td>
<td><strong>0.75</strong></td>
</tr>
<tr>
<td>S₂</td>
<td>C. elegans</td>
<td>iNuc-PseKNC</td>
<td>86.90</td>
<td>90.30</td>
<td>83.55</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iNuc-STNC</td>
<td><strong>88.62</strong></td>
<td><strong>91.62</strong></td>
<td><strong>86.66</strong></td>
<td><strong>0.77</strong></td>
</tr>
<tr>
<td>S₃</td>
<td>D. melanogaster</td>
<td>iNuc-PseKNC</td>
<td>79.77</td>
<td>78.31</td>
<td>81.65</td>
<td>0.60</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iNuc-STNC</td>
<td><strong>81.67</strong></td>
<td><strong>79.76</strong></td>
<td><strong>83.61</strong></td>
<td><strong>0.63</strong></td>
</tr>
</tbody>
</table>
5.1 Introduction

In the last chapter, the first phase of research was comprehensively discussed. This chapter commences with the second phase of research. The second phase explains an automated model for identification of transmembrane protein structures. Owing to the essential role of transmembrane proteins in living species, the identification of transmembrane proteins is inevitable. However, all the important knowledge about the functions and structures of transmembrane proteins are reflected from transmembrane topology. In spite of that, only 1% of transmembrane proteins structure are available. Owing to a limited number of recognized structures and experimental complexity, the identification of transmembrane helix and topology becomes a major problem in bioinformatics and proteomics. In last few decades, sequential information of amino acids was utilized for recognition of transmembrane helices location and orientation instead of structure information. Looking at the vital role of sequential information in identification of transmembrane helix, we have proposed a sequence based prediction model PSOFuzzySVM-TMH in order to correctly identify the location of transmembrane helix. The protein sequences were formulated by evolutionary profiles based method position specific scoring matrix (PSSM) and physiochemical properties of amino acids based method 6-letter exchange group in order to exploit all the salient, pronounced and variant numerical descriptors. Sometimes, feature space has irrelevant, noisy and repetitive information, consequently, misclassification, difficulty in clear pattern discerning and high dimensionality are the limitations of unnecessary features. To reduce the extraneous information along with enhancing the learning capability of prediction model, evolutionary intelligent feature selection method such as particle swarm optimization (PSO) was employed. Afterward, merged the selected feature spaces of the 6-letter exchange group and PSSM by making a hybrid feature space. For
learning hypothesis, Fuzzy SVM is used where the concept of Fuzzy was incorporated with simple SVM. In this study, 10-fold cross validation test is applied for the assessment of PSOFuzzySVM-TMH model at different levels i.e., per segment, per protein, and per residue, using two different benchmark datasets.

5.2 Materials and Methods

In this section, discussion regarding various datasets was carried out, followed feature extraction techniques, and presented our proposed PSOFuzzySVM-TMH model.

5.2.1. Datasets

In this work, two different benchmark datasets were considered. The first benchmark dataset-1 given by Moller et al., which contains a low-resolution transmembrane protein sequences [187]. It is derived from SWISS-PROT release 49.0.41 [188]. Firstly, the dataset-1 composed of 145 different protein sequences. Latterly, two sequences were excluded because they have lacking annotation with transmembrane proteins. Consequently, dataset-1 comprising 143 sequences having 687 transmembrane helix segments.

The second benchmark dataset-2 contains a high-resolution transmembrane protein sequences. In dataset-2, 101 transmembrane protein sequences of 3D helix structure are collected from MPtopo database [189], while 231 transmembrane protein sequences are selected from TMPDB database [163]. By merging the two datasets, 30% CD-HIT software is applied to minimize the similarity and homologous. Finally, dataset-2 including 258 single and multi-spanning transmembrane protein sequences having 1232 transmembrane helix segments.

5.2.2. Feature Extraction Techniques

The numerical descriptors are extracted from protein sequences using two different feature extraction techniques i.e., evolutionary profiles based method PSSM and physiochemical properties of amino acids based method 6-letter exchange group. These feature extraction techniques were discussed in detail in Chapter 3.
5.2.3. Proposed PSOFuzzySVM-TMH Prediction Model

In this work, an efficient and accurate PSOFuzzySVM-TMH model was developed for the identification of transmembrane helix segments. The PSOFuzzySVM-TMH model has two distinct feature extraction methods: 6-letter exchange group and PSSM are utilized for representation of protein sequences. The structure of some amino acids is similar because of its nature. On the basis of these similarities, the amino acids are distributed into six distinct groups known as 6-letter exchange group representation. In this method, initially the amino acids are substituted by the corresponding group, which is illustrated in (Chapter No 3, Table 3.1), was obtained using PAM matrix [190]. For instance, all amino acids i-e R, K, and H in the novel sequence are substituted by a₁, E, Q, D, and N are substituted by a₂, and C is replaced by a₃ and so forth. After substituting the whole residues of an original sequence by 6-letter namely; a₁, a₂, a₃, a₄, a₅, and a₆, finally, the resultant sequence consists specifically these 6 various characters. Secondly, various size of sliding windows are applied, consequently, 6 numerical descriptors are extracted from every sequence position and then the window is shifted to the next position of the protein sequence. This process is reiterated till the last residue of the sequence.

By applying the second feature extraction method PSSM, in this method protein sequences are executed by PSI-Blast tool. As a result, for each residue of a protein sequence, 20 values are generated, which determining the fractions of mutations detected at the specific position in a protein family. After that, by applying sliding window and centered on a target residue with 4 residues on each side of the target residue. Finally, 180-D feature space is generated. To select highly variant and salient features and also removing the repetitive as well as irrelevant features, PSO is applied as an evolutionary intelligent feature selection technique on each feature space independently. Consequently, 4D features are selected from 6-letter exchange group representation and 90D features are selected from PSSM feature space.

These selected feature spaces are merged to produce a hybrid feature space, accordingly hybrid feature space has the dimension of 94D [191, 192]. Fuzzy SVM is utilized as a learning hypothesis. The proposed framework of the prediction PSOFuzzySVM-TMH model is shown in Figure 5.1.
In order to validate the proposed predicted model i.e., PSOFuzzySVM-TMH, it is evaluated and measured at three different levels i.e., per segment, per protein and per residue basis. The proposed model is examined on the basis of these measures namely: accuracy, recall, precision, and MCC.

\[
Q_{htm}^{\%\text{obsd}} = \left( \frac{\text{Number of correctly predicted transmembrane helix in dataset}}{\text{Total number of transmembrane helix in dataset}} \right) \times 100 \tag{5.1}
\]

where \( Q_{htm}^{\%\text{obsd}} \) show the recall of transmembrane helix segments.

\[
Q_{htm}^{\%\text{prd}} = \left( \frac{\text{Number of correctly predicted transmembrane helix in dataset}}{\text{Number of transmembrane helix predicted in dataset}} \right) \times 100 \tag{5.2}
\]
where $Q_{\text{err}}^{\text{pred}}$ show the precision of transmembrane helix segments.

$$Q_{\text{ok}} = \left( \frac{\sum_{i} \delta_{i}}{N_{\text{Prot}}} \right) \times 100 \quad \delta_{i} = \begin{cases} 1, & \text{if } Q_{\text{hm}}^{\text{pred}} = 100 \text{ for protein } i \\ 0, & \text{otherwise} \end{cases}$$ (5.3)

where $Q_{\text{ok}}$ show the accuracy at protein level having all its transmembrane helix segments are correctly identified.

$$Q_{2} = \left( \sum_{i} \left( \frac{(\text{Number of residues correctly predicted in protein } i)}{(\text{Number of residues in protein } i)} \right) \right) \times 100$$ (5.4)

where $Q_{2}$ represents the percentage of residues present in the transmembrane helix and non-transmembrane helix segments are predicted accurately.

$$Q_{2T}^{\text{globul}} = \left( \frac{\text{Number of residues correctly predicted in transmembrane helices}}{\text{Number of residues observed in transmembrane helices}} \right) \times 100$$ (5.5)

where $Q_{2T}^{\text{globul}}$ describes the number of residues that are correctly predicted in the observed residues.

$$Q_{2T}^{\text{pred}} = \left( \frac{\text{Number of residues correctly predicted in transmembrane helices}}{\text{Number of residues predicted in transmembrane helices}} \right) \times 100$$ (5.6)

where $Q_{2T}^{\text{pred}}$ measures the number of residues that are predicted correctly in the predicted residues.

$$MCC = \left( \frac{(TN \times TP) - (FP \times FN)}{\sqrt((FN + TN)(TP + FN)(FP + TN)(FP + TP))} \right)$$ (5.7)

Where

- TP denotes the number of transmembrane helix residues which are predicted correctly.
- FP shows the number of transmembrane helix residues which are predicted incorrectly.
- TN and FN represent the number of non-transmembrane helix residues which are correctly and incorrectly predicted, respectively.

5.3 Results and Discussion

In this work, a statistical test 10-fold cross-validation test is applied in connection to minimize the execution cost of jackknife test. The training of learning hypothesis is conducted on 9/10 folds whereas the test is performed 1/10 fold. This mechanism is executed 10 times to allow the learning hypothesis towards each fold for testing. The performances of the PSOFuzzySVM-TMH model having selected feature space and full feature space along with their hybrid space are shown in the sub section followed. The performance of the PSOFuzzySVM-TMH model is examined at three different levels such as per segment, per protein, and per residue.

5.3.1. Performance analysis of PSOFuzzySVM-TMH on PSSM feature space

Table 5.1 presented the experimental results of PSOFuzzySVM-TMH model on PSSM based complete and selected feature spaces. By analyzing the first low-resolution dataset, the model achieved 67.8% of accuracy at per protein level, whereas it has yielded 93.6% and 94.3% of recall and precision at segment level. At per residue level, the PSOFuzzySVM-TMH model achieved 88.0% accuracy, whereas, the values of precision, recall and MCC are 87.2%, 79.2%, and 0.77, respectively. For high-resolution dataset, it has obtained 70.1% of accuracy at protein level, whereas it has achieved 96.1% of precision, and 95.2% of recall at segment level. Similarly, the obtained results at per residue level are 90.9% of accuracy, 86.7% of precision, 91.4% of recall, and 0.82 of MCC.

The evolutionary intelligent feature selection technique namely PSO is applied to enhance the discrimination and generalization power of learning hypothesis, it is used to select high discriminative features from full feature space. Further, the result of the selected feature space on low resolution dataset is at protein level, it has achieved
an accuracy of 71.3%, at segment level the proposed model yielded the value of recall and precision are 95.3% and 94.6%, respectively, whereas, at residue level it has obtained 89.5% of accuracy, 88.9% of precision, 81.2% of recall and 0.78 of MCC. Likewise, for high-resolution dataset, PSOFuzzySVM-TMH achieved 72.6% accuracy at protein level, whereas at segment level obtained 97.0% precision and 96.7% recall. Similarly, at residue level, the predicted accuracy, precision, recall, MCC are 92.0%, 88.4%, 92.6%, and 0.83, respectively.

Table 5.1 Performance analysis of PSSM feature space at different levels

<table>
<thead>
<tr>
<th>Feature space</th>
<th>Per Segments</th>
<th>Per Proteins</th>
<th>Per Residue</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Q\text{obsd}</td>
<td>Q\text{prd}</td>
<td>Q_{ok}</td>
</tr>
<tr>
<td>Low resolution</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selected feature space</td>
<td>94.6</td>
<td>95.3</td>
<td>71.3</td>
</tr>
<tr>
<td>Full feature space</td>
<td>94.3</td>
<td>93.6</td>
<td>67.8</td>
</tr>
<tr>
<td>High resolution</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selected feature space</td>
<td>97.0</td>
<td>96.7</td>
<td>72.6</td>
</tr>
<tr>
<td>Full feature space</td>
<td>96.1</td>
<td>95.2</td>
<td>70.1</td>
</tr>
</tbody>
</table>

Figure 5.2 Performance of PSSM feature space for low resolution dataset.
5.3.2. Performance analysis of PSOFuzzySVM-TMH on 6-letter exchange group

Table 5.2 illustrates the success rate of proposed PSOFuzzySVM-TMH model on 6-letter exchange group based full and selected feature spaces. For low-resolution dataset at protein level, the model yielded 69.2% of accuracy, while at segment level, it yielded 94.1% recall and 94.7% precision values. Similarly, at per residue level, it has obtained 88.3% accuracy, 87.9% precision, 80.2% recall, and 0.77 MCC. In contrast, at protein level the accuracy of proposed prediction PSOFuzzySVM-TMH model is 70.1%, for high resolution dataset. Further, at segment level the value of precision is 95.2% and recall is 96.0% whereas the proposed model yielded an accuracy, precision, recall, and MCC is 90.2%, 86.9%, 91.8%, and 0.81, respectively at residue level.

The result of the selected feature space on low resolution dataset is at protein level, it has achieved an accuracy of 72.0%, at segment level, the proposed model yielded the value of recall and precision are 95.8% and 95.2%, whereas, at residue level,
it has obtained 89.1% of accuracy, 88.3% of precision, 81.0% of recall and 0.78 of MCC.

On the other hand, PSOFuzzySVM-TMH model achieved 73.9% accuracy for high resolution dataset at protein level, further; at segment level it has obtained 96.7% of precision and 97.3% of recall. At residue level the prediction model obtained the value of accuracy, precision, recall, and MCC is 91.9%, 88.0%, 92.9%, and 0.82, respectively.

Table 5.2 Performance analysis of 6-letter exchange group feature spaces at different levels

<table>
<thead>
<tr>
<th>Feature space</th>
<th>Per Segments</th>
<th>Per Proteins</th>
<th>Per Residue</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Q&lt;br&gt;obsd</td>
<td>Q&lt;br&gt;prd</td>
<td>Q&lt;br&gt;ok</td>
</tr>
<tr>
<td>Low resolution</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selected feature space</td>
<td>95.2</td>
<td>95.8</td>
<td>72.0</td>
</tr>
<tr>
<td>Full feature space</td>
<td>94.7</td>
<td>94.1</td>
<td>69.2</td>
</tr>
<tr>
<td>High resolution</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selected feature space</td>
<td>96.7</td>
<td>97.3</td>
<td>73.9</td>
</tr>
<tr>
<td>Full feature space</td>
<td>95.2</td>
<td>96.0</td>
<td>70.1</td>
</tr>
</tbody>
</table>

Figure 5.4 Performance of 6-letter exchange group feature spaces for low resolution dataset.
5.3.3. Performance analysis of PSOFuzzySVM-TMH on Hybrid feature space

To improve the discrimination power of PSOFuzzySVM-TMH model, both the full and selected feature spaces of PSSM and 6-letter exchange group are merged by sum rule in order to form hybrid space. Table 5.3 reported the of proposed prediction PSOFuzzySVM-TMH model. By analyzing the performance of hybrid space of full feature spaces for low-resolution dataset, the success rates of PSOFuzzySVM-TMH model obtained an accuracy of 75.5% at protein level, whereas it has yielded 95.7% recall and 95.6% precision values at segment level. Similarly at per residue level it has achieved 90.7% accuracy, further, the values of precision, recall and MCC are 89.1%, 83.4%, and 0.79, respectively. In contrast, the accuracy of the model at protein level is 77.5%, for high resolution dataset, whereas it has yielded 96.6% of precision and 96.3% of recall at segment level. Similarly at residue level, it has obtained values of accuracy, precision, recall, and MCC is 92.5%, 80.3%, 93.2%, and 0.84, respectively.
By examining the performance of proposed model on hybrid space of selected feature spaces. The predicted outcome of PSOFuzzySVM-TMH model is listed in Table 5.3. The success rates for low resolution dataset, it has accuracy at protein level are 77.6%. Whereas the performance at segment level, the proposed model yielded the value of recall and precision are 97.1% and 97.0%, whereas, at residue level obtained 93.8% of accuracy, 91.8% of precision, 85.1% of recall and 0.81 of MCC. On the other hand, PSOFuzzySVM-TMH model achieved 79.3% accuracy for high resolution dataset at protein level, further, at segment level, the results are yielded for precision and recall is 97.5% and 98.2%. At residue level, the prediction model obtained the value of accuracy, precision, recall, MCC is 94.6%, 92.8%, 95.7%, and 0.86, respectively.

After empirical analysis, it is concluded that the performance of selected feature spaces is considered efficient compared to un-selected feature spaces. Besides, the prediction performance of PSOFuzzySVM-TMH is sound using 6-letter exchange group in case of individual feature space. On the other hand, the prediction performance of PSOFuzzySVM-TMH with hybrid feature space is quite encouraging compared to individual feature spaces because hybrid feature space reflects the discriminative power of the two different feature spaces. Furthermore, the success rates of PSOFuzzySVM-TMH for high resolution dataset are more efficient compared to low resolution dataset. The low resolution dataset contains some main issues such as signal peptides are not removed from some low resolution TM proteins and low reliability annotation proteins.

<p>| Table 5.3 Performance analysis of Hybrid feature space at different levels |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Feature space   | Per Segments    | Per Proteins    | Per Residue     |                |                |                |                |</p>
<table>
<thead>
<tr>
<th></th>
<th>Q^{obsd}</th>
<th>Q^{prd}</th>
<th>Q_{ok}</th>
<th>Q_{2}</th>
<th>Q^{obsd}</th>
<th>Q^{prd}</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low resolution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selected feature space</td>
<td>97.0</td>
<td>97.1</td>
<td>77.6</td>
<td>93.8</td>
<td>91.8</td>
<td>85.1</td>
<td>0.81</td>
</tr>
<tr>
<td>Full feature space</td>
<td>95.6</td>
<td>95.7</td>
<td>75.5</td>
<td>93.8</td>
<td>91.8</td>
<td>85.1</td>
<td>0.81</td>
</tr>
<tr>
<td>High resolution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selected feature space</td>
<td>97.5</td>
<td>98.2</td>
<td>79.3</td>
<td>94.6</td>
<td>92.8</td>
<td>95.7</td>
<td>0.86</td>
</tr>
<tr>
<td>Full feature space</td>
<td>96.9</td>
<td>96.3</td>
<td>77.5</td>
<td>92.5</td>
<td>90.3</td>
<td>93.2</td>
<td>0.84</td>
</tr>
</tbody>
</table>
Figure 5.6 Performance of Hybrid feature spaces for low resolution dataset

Figure 5.7 Performance of Hybrid feature spaces for high resolution dataset
5.3.4. Performance comparison of PSOFuzzySVM-TMH model with existing models

The proposed model is not only compared with implemented models but also compared with existing models in the literature. The comparison has been brought at various levels: at protein, at segment and at residue levels.

In Table 5.4, success rates of proposed PSOFuzzySVM-TMH model have been compared with existing models at various levels to demonstrate the strength of proposed model. By analyzing the success rate of proposed model at low-resolution dataset, the PSOFuzzySVM-TMH model achieved the 77.61% of accuracy compared to existing models. In the present state of the art methodologies, Lo et al., proposed SVMtop model obtaining the 73.29% accuracy. Similarly, Arai et al., developed model having 74.83% of accuracy. Likewise, the performance of PSOFuzzySVM-TMH model is also evaluated with other existing models namely: PHDhtm v.1.96, SOSUI 1.1, MEMSAT3, HMMTOP2, Phobius, TMHMM2, SPLIT4, and Top-Pred2. The performance of PSOFuzzySVM-TMH model at the segment level is estimated using two performance measure parameters i.e., recall and precision. The PSOFuzzySVM-TMH model achieved 97.07% precision and 97.12% recall whereas the existing model, SVMtop has obtained 93.94% of precision and 94.76% of recall. Similarly, the performance at per residue level, the PSOFuzzySVM-TMH model is measured by using various metrics namely; precision, accuracy, recall, and MCC. The proposed model has achieved 85.15% precision, 93.81% accuracy, 91.82% recall, and 0.81 MCC. In contrast, the predicted results of existing SVMtop model were 80.35% precision, 89.23% accuracy, 87.50% recall, and 0.77 MCC. Furthermore, by examining the proposed model in terms of performance with that of existing models at a high resolution dataset, the PSOFuzzySVM-TMH still obtained the highest accuracy 79.32%. Whereas the existing model SVMtop model has obtained 72.09% accuracy at per protein level. The precision and recall of PSOFuzzySVM-TMH model are 97.57% and 98.21% at segment level. Similarly at the residue level, the performance of PSOFuzzySVM-TMH model are 95.73% precision, 94.13% accuracy, 92.82% recall, and 0.86 MCC, while the predicted results of SVMtop model were 84.36% precision, 90.90% accuracy, 87.84% recall, and 0.81 MCC.
<table>
<thead>
<tr>
<th>Per segment (%)</th>
<th>Per Protein (%)</th>
<th>Per residue (%)</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q^\text{obsd}$</td>
<td>$Q^\text{ok}$</td>
<td>$Q^\text{sprd}$</td>
<td>$Q^2$</td>
</tr>
<tr>
<td>Low resolution</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSOFuzzySVM-TMH</td>
<td>97.07</td>
<td>74.20</td>
<td>97.12</td>
</tr>
<tr>
<td>SVMtop</td>
<td>94.76</td>
<td>73.29</td>
<td>93.94</td>
</tr>
<tr>
<td>SPLIT4</td>
<td>93.45</td>
<td>72.73</td>
<td>91.32</td>
</tr>
<tr>
<td>ConPred-II</td>
<td>94.76</td>
<td>74.83</td>
<td>92.21</td>
</tr>
<tr>
<td>Phobius</td>
<td>92.87</td>
<td>72.03</td>
<td>93.14</td>
</tr>
<tr>
<td>PolyPhobius</td>
<td>94.47</td>
<td>71.33</td>
<td>91.54</td>
</tr>
<tr>
<td>TMHMM2</td>
<td>90.39</td>
<td>68.53</td>
<td>93.52</td>
</tr>
<tr>
<td>HMMTOP2</td>
<td>89.96</td>
<td>64.34</td>
<td>93.78</td>
</tr>
<tr>
<td>PHDhtm v.1.96</td>
<td>76.27</td>
<td>39.86</td>
<td>85.76</td>
</tr>
<tr>
<td>MEMSAT3</td>
<td>91.56</td>
<td>70.63</td>
<td>90.24</td>
</tr>
<tr>
<td>TopPred2</td>
<td>86.75</td>
<td>57.34</td>
<td>91.13</td>
</tr>
<tr>
<td>SOSUI 1.1</td>
<td>88.36</td>
<td>63.64</td>
<td>91.55</td>
</tr>
<tr>
<td>High resolution</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSOFuzzySVM-TMH</td>
<td>97.57</td>
<td>79.32</td>
<td>98.21</td>
</tr>
<tr>
<td>SVMtop</td>
<td>92.78</td>
<td>72.09</td>
<td>94.46</td>
</tr>
<tr>
<td>SPLIT4</td>
<td>89.77</td>
<td>65.12</td>
<td>91.56</td>
</tr>
<tr>
<td>ConPred-II</td>
<td>90.94</td>
<td>69.14</td>
<td>91.31</td>
</tr>
<tr>
<td>Phobius</td>
<td>88.72</td>
<td>67.05</td>
<td>93.58</td>
</tr>
<tr>
<td>PolyPhobius</td>
<td>90.91</td>
<td>67.44</td>
<td>91.28</td>
</tr>
<tr>
<td>TMHMM2</td>
<td>86.93</td>
<td>59.30</td>
<td>93.78</td>
</tr>
<tr>
<td>HMMTOP2</td>
<td>90.34</td>
<td>65.89</td>
<td>89.98</td>
</tr>
<tr>
<td>PHDhtm v.1.96</td>
<td>74.43</td>
<td>38.37</td>
<td>84.59</td>
</tr>
<tr>
<td>MEMSAT3</td>
<td>87.67</td>
<td>64.84</td>
<td>91.09</td>
</tr>
<tr>
<td>TopPred2</td>
<td>84.50</td>
<td>50.39</td>
<td>90.05</td>
</tr>
<tr>
<td>SOSUI 1.1</td>
<td>85.06</td>
<td>56.98</td>
<td>92.17</td>
</tr>
</tbody>
</table>

After analyzing the experimental results, it is concluded that the classification rates of our proposed PSOFuzzySVM-TMH model are quite encouraging at each level in both the datasets. These major improvements in terms of all using measures have been
credited to the amalgamation of two powerful and informative formulation schemes, the selection of valuable features through evolutionary intelligent feature selection technique, and the best learning hypothesis.
Chapter 6

CONCLUSIONS AND FUTURE DIRECTIONS

Owing to the rudimentary roles of the nucleosome and transmembrane proteins in living species, an effort has been carried out in development of sequence-based predictors or models for nucleosome positioning in genomes and transmembrane proteins. These models contribute in the area of proteomics, bioinformatics, and genomics by applying different contemporary intelligent techniques to increase the classification rates based on biological sequences. The whole work was accomplished in two phases. In the first phase of the thesis, DNA was targeted where the nucleosome positioning in genomes was identified with high precision. On the other hand, the second phase of thesis focused on protein by predicting transmembrane helices. The biological sequences were formulated by discrete, evolutionary profiles and physicochemical properties of amino acids based methods in order to truly reflect the target classes. Modern and intelligent Machine learning algorithms were applied in order to predict nucleosome positioning and transmembrane protein more accurately and efficiently. Jackknife and 10-fold cross validation tests were employed to calculate the performance of the learning hypotheses. The performance is computed on the basis of various metrics namely: specificity, sensitivity, accuracy, MCC, recall, precision, and F-measure.

6.1 Nucleosome Positioning in Genomes

In chapter 4, we introduced the first phase of our research. In this phase, we developed a computational model i.e., iNuc-STNC for the identification of nucleosome positioning in genomes. Dinucleotide composition (DNC), split trinucleotide composition (STNC) and trinucleotide composition (TNC) are DNA sequences representation methods, which were adopted to extract nominal values. Then, these extracted numerical values were provided to three distinct learning hypotheses i.e., SVM, KNN, and PNN. The maximum predicted results of these learning hypotheses were examined and noted down. It was observed the success rates of SVM in combination with STNC feature space was quite encouraging and outstanding not only
among other learning hypotheses and feature spaces but also from existing methods in the literature so far.

6.2 Transmembrane Proteins

In chapter 5, we presented the second phase of our research. In this phase, we developed a computational model i.e., PSOFuzzySVM-TMH for the identification of transmembrane helix segments. In this model, two feature spaces were used such as an evolutionary profile based method position specific scoring matrix (PSSM) and physicochemical properties of amino acids based method 6-letter exchange group in order to exploit all the salient, pronounced and variant numerical descriptors. Sometimes, feature space has irrelevant, noisy and repetitive information, consequently, misclassification, difficulty in clear pattern discerning and high dimensionality are the limitations of unnecessary features. In order to reduce the extraneous information along with enhancing the learning capability of prediction model, evolutionary intelligent feature selection technique particle swarm optimization (PSO) was applied. After that, the selected feature spaces of the 6-letter exchange group and PSSM are further merged to form a hybrid space. Fuzzy SVM is used as learning hypothesis, where the concept of Fuzzy was incorporated with simple SVM.

Finally, we have concluded that our proposed models for nucleosome positioning in genomes and transmembrane proteins might play a significant role not only in Molecular Biology, Computational Biology, and Bioinformatics, but also in pharmaceutical industries.

6.3 Future Directions

Due to the huge amount of DNA and proteins biological sequences generated and added to data banks, it is a big challenge for researchers to accurately identify nucleosome positioning in genomes and transmembrane helix segments. In this regards, tremendous efforts have been carried out and sort out a lot of problems which were facing in traditional approaches. Various user friendly online web predictors were launched. But still, space for improvement exists in term of space, time and high success rates. In this study, several feature spaces and computational models have been proposed to identify
nucleosome positioning in genomes and transmembrane helix segments in proteins with high accuracy. In future,

- To make efforts to improve the performance of these prediction models further.
- To develop web predictors, which are freely available to the research community.
- To reduce the computational complexity and real time cost of these predictors.
References


78


[58] M. Schiffer, A.B. Edmundson, Use of helical wheels to represent the structures of proteins and to identify segments with helical potential, Biophysical Journal, 7 (1967) 121.


