COMPUTATIONAL ANALYSIS OF PROTEIN STRUCTURE AND ITS SUBCELLULAR LOCALIZATION USING AMINO ACID SEQUENCES

Thesis
Submitted in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

Mr. Sanjay S. Bankapur

DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA
SURATHKAL, MANGALORE - 575025

February 2021

Declaration

I hereby *declare* that the Research Thesis entitled “Computational Analysis of Protein Structure and its Subcellular Localization using Amino Acid Sequences” which is being submitted to the National Institute of Technology Karnataka, Surathkal in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy in Information Technology is a *bonafide report of the research work carried out by me*. The material contained in this thesis has not been submitted to any University or Institution for the award of any degree.

Sanjay B.S (22|FEB|2021)

Mr. Sanjay S. Bankapur
Register No.: 148046-IT14F05
Department of Information Technology

Place: NITK Surathkal

Date: 22-Feb-2021

Certificate

This is to *certify* that the Research Thesis entitled "Computational Analysis of Protein Structure and its Subcellular Localization using Amino Acid Sequences" submitted by Mr. Sanjay S. Bankapur (Register Number: 148046-IT14F05) as the record of the research work carried out by him, is *accepted as the Research Thesis submission* in partial fulfilment of the requirements for the award of degree of Doctor of Philosophy.

Dr. Nagamma Patil
Research Guide
Assistant Professor
Department of Information Technology
NITK Surathkal - 575025

Chairman - DRPC
(Signature with Date and Seal)

Acknowledgements

First and foremost, I would like to thank my mother for constant support, inspiration, encouragement, unconditional cooperation, love and emotional support during my research work. Also, I would like to thank my elder brother, sister-in-law, younger sister, brother-in-law, nephew, and my father for their unconditional cooperation and emotional support.

I express sincere gratitude to my research guide, Dr. Nagamma Patil and members of my RPAC committee, Prof. Ananthanarayana V. S. and Dr. Pushparaj Shetty D. for their valuable feedback and constructive suggestions during my research work.

I sincerely thank the Ministry of Human Resource Development, Government of India for sponsoring my research fellowship. I also extend my heartfelt thanks for Vision Group on Science and Technology, Department of Information Technology, Biotechnology and Science & Technology, Govt. of Karnataka, India for funding this research project.

I would like to thank Prof. Robert D'Souza (NITK, Surathkal), Dr. Santhosh Onkar, and Dr. Balaji S. (MIT, Manipal) for mentoring me and providing feasible inputs.

I express my heartfelt thanks to my friends Mr. Ravishankar S. G., Mr. Nrupatunga Y., Mr. Praveen Kumar R., Mr. Sri Harsha S. P., Mr. Sharath N. J, Mr. Chethan A. G. and Mr. Kumar Sandeep for their constant motivation, encouragement, and continuous moral support throughout my research span.

I would like to extend my sincere gratitude to Dr. Manjunath K. Vanahalli, Dr. Gokul S. Krishnan, Dr. Girish G. N., Dr. Shridhar G. Domanal and my co-researchers who supported me all along and made my Ph.D. journey more memorable.

I express my sincere thanks to Department of Information Technology, all my fellow doctoral students, teaching and non-teaching staffs of Department of Information Technology for their continuous support and encouragement.

(Mr. Sanjay S. Bankapur)

Abstract

A cell is the basic unit of all organisms. In a cellular life cycle, various complex metabolic activities are being carried out in different cell compartments. Protein plays an important role in many complex metabolic activities. Proteins are generated in the post-transcriptional modification activity of a cell. Initially, the generated proteins are in the linear structure and it is called as protein primary structure. Within the cell, proteins tend to move from one compartment (subcellular location) to other compartments, and based on the environment (in which the primarily structured proteins reside), primary structured proteins transform into secondary and tertiary structures. Tertiary structured proteins interact with nearby structured proteins to form a quaternary structure. A protein performs its biological functions when it attains its respective tertiary structure.

Identification of a protein structure and its subcellular locations are challenging and important tasks in the field of medical science. Various health issues are identified and solved via novel drug discoveries and a prior and accurate knowledge of protein structure and its subcellular location helps in developing a respective drug. In order to identify protein structure and its subcellular locations, various biological methods such as X-ray crystallography, nuclear magnetic resonance spectroscopy, cell fractionation, fluorescence microscopy, and electron microscopy are used. The main advantage of biological methods is that they are accurate in identifying protein structures and its subcellular locations. The disadvantages of biological methods are that they are time-consuming and very expensive. In this post-genomic era, high-volumes of protein primary structures are decoded by various research communities and are added to protein data banks. Identification of protein structure and its subcellular locations using biological methods are not a feasible option for high-volumes of proteins.

Over the decades, various computational methods have been proposed to identify protein structure and its locations; however, the existing computational methods exhibit limited accuracy and hence they are less effective. The main objective of this thesis is to propose effective computational models that contribute to the prediction of protein structure and its subcellular locations. In this regard, four important and specific problems of protein structure and its subcellular location have been solved and they are: (i) multiple sequence alignment, (ii) protein secondary structural class prediction, (iii) protein fold recognition, and (iv) protein subcellular localization prediction.

The importance of multiple sequence alignment is that a vital and consistent homologous pattern of proteins can be captured and these patterns will further help in solving protein structure and its subcellular locations. The proposed alignment method includes three main modules: a) an effective scoring system to score the quality of the aligned sequences, b) a progressive-based alignment approach is adopted and modified to align multiple sequences, and c) the aligned sequences are refined using the proposed polynomial-time complexity-based single iterative optimization framework. The proposed method has been assessed on publicly available benchmark datasets and recorded 17.7% improvement over the CLUSTAL X model on the BAliBASE dataset.

Identification of protein secondary structural class is one of the important tasks that further help in the prediction of protein tertiary structure. Protein secondary structural class prediction is a supervised problem that falls under the multi-class category. The proposed protein secondary structural class prediction model contains a novel feature modelling strategy that extracts global and local features followed by a novel ensemble of classifiers to predict structural class. The proposed model has been assessed on both publicly available benchmark datasets and derived latest high-volume datasets. The performance of the proposed model recorded an improvement of 5.3% on the 25PDB dataset over one of the best predictors from the literature.

A protein fold recognition is a categorization of various folds of a protein that exhibits in tertiary structure. Protein fold recognition is a supervised problem that falls under the multi-class category. The proposed fold recognition model contains a novel and effective feature modelling approach that includes Convolutional and SkipXGram bi-gram techniques to extract global and local features followed by an effective deep learning framework for fold recognition. The proposed model has been assessed on both publicly available benchmark datasets and derived latest high-volume datasets. The performance of the proposed model recorded a relative improvement of 5% on the DD dataset over one of the best predictors from the literature.

An effective protein sub-chloroplast localization prediction model is proposed to solve one-level more microscopic problem of subcellular localization. Protein sub-chloroplast localization is a supervised problem that falls under the multi-class and multi-label category. The proposed protein sub-chloroplast localization prediction model contains a novel feature extraction technique such as SkipXGram bi-gram followed by a deep learning framework for multi-label classification. The proposed model has been assessed on publicly available benchmark datasets and recorded an improvement of (absolute) 30.39% on the Novel dataset over the best predictor from the literature.

Keywords: Progressive alignment; Look back ahead scoring strategy; Position-residue specific dynamic gap penalty scoring strategy; Single iterative optimization; Embedding; Skip-gram bi-gram; Evolutionary profiles; Ensemble classifier; Deep learning; Binary Relevance; Genetic Algorithm; Machine Learning

Contents

1 Introduction 1

1.1 Research Motivation 4
1.2 Research Challenges 4
1.3 Major Contributions of the Thesis 5
1.4 Organization of the Thesis 6
1.5 Summary 7

2 Literature Survey 9

2.1 Multiple Protein Sequence Alignment 9
2.2 Protein Secondary Structural Class 15
2.3 Protein Fold Recognition 20
2.4 Protein Subcellular Localization 24
2.5 Research Gaps 28
2.6 Problem Statement 29
2.7 Research Objectives 30
2.8 Summary 30

3 Multiple Protein Sequence Alignment 31

3.1 General Problem Statement: 31
3.2 Scoring System 31
  3.2.1 Look Back Ahead Scoring Strategy 32
  3.2.2 Position-Residue Specific Dynamic Gap Penalty Scoring Strategy 34
3.3 Progressive Alignment Method using LBA and PRSDGP Scoring Strategies 35
3.4 Single Iterative Optimization Framework to improve Local Optima . 36
  3.4.1 Total Time Complexity Analysis of the Proposed Model . . 40
3.5 Results and Discussion 40
  3.5.1 Evaluation Metrics 41
  3.5.2 Experiment Setup 43
  3.5.3 Results and Analysis 43

3.6 Summary 50

4 Protein Secondary Structural Class Prediction 53

4.1 Preamble 53
4.2 Datasets 54
  4.2.1 Data Preparation 54
4.3 PSSC Prediction using Local Features with State-of-the-art Classifiers 56
  4.3.1 Datasets 56
  4.3.2 Feature Extraction 56
    4.3.2.1 SkipXGram bi-gram (SXGbg) Technique 56
    4.3.2.2 Character Embedding (CE) Technique 57
  4.3.3 Classification 58
  4.3.4 Proposed Model 59
  4.3.5 Results and Discussion 60
    4.3.5.1 Environment Setup and Performance Measure . . 60
    4.3.5.2 Performance Analysis on Character Embedding Features 60
    4.3.5.3 Performance Analysis on SkipXGram bigram (SXGbg) Features 61
    4.3.5.4 Performance Analysis on the Proposed Feature Model using State-of-the-art Classifiers 62
    4.3.5.5 Performance Analysis of the Proposed Model against State-of-the-art Models 62
4.4 PSSC Prediction using Local and Global Features with Ensemble of Classifiers 65
  4.4.1 Datasets 65
  4.4.2 Data Preparations 65
  4.4.3 Feature Modeling 65
    4.4.3.1 Character Embedding (CE) Technique 66
    4.4.3.2 SkipXGram bi-gram (SXGbg) Technique 67
    4.4.3.3 General Statistical (GS) based Feature 67
  4.4.4 Classification 68

4.4.5 Proposed Model 69
4.4.6 Results and Discussion 69
4.4.6.1 Experimental Setup and Performance Measures . 71
4.4.6.2 Performance Analysis of Embedding based Feature Sets 71
4.4.6.3 Performance Analysis of SkipXGram based Feature Sets 73
4.4.6.4 Performance Analysis of the Proposed Feature Extraction Techniques 74
4.4.6.5 Performance Analysis of the Proposed Model with State-of-the-art Models 75
4.4.6.6 Performance Analysis of the Proposed Model on Large-Scale Updated Dataset 79
4.4.6.7 Statistical Significance Analysis 80
4.5 Summary 80

5 Protein Fold Recognition 83

5.1 Datasets 83
5.1.1 Benchmark Datasets 83
5.1.2 Latest High-volume Low-similarity Datasets 84
5.1.3 Data Preparations 84
5.2 Proposed Methodology: 86
5.2.1 Feature Extraction: 86
5.2.1.1 Convolutional (Conv) Features 86
5.2.1.2 SkipXGram bi-gram (SXGbg) Features 87
5.2.2 Deep Neural Network: 88
5.2.3 The Proposed Model: 89
5.2.3.1 Parameter Optimization: 90
5.3 Results and Discussions 90
5.3.1 Experimental Setup 90
5.3.2 An ablation study of feature sets on the derived datasets of SCOPe 2.07 91

5.3.3 Comparison with state-of-the-art models: 93
5.3.4 Statistical Significance Analysis: 95
5.4 Summary 96

6 Protein Subcellular Localization Prediction 97
6.1 Datasets 97
6.2 Data Preparations 98
6.3 Protein Sub-Chloroplast Localization Prediction using Binary Relevance 98
6.3.1 Feature Modeling 98
6.3.1.1 Evolutionary Profile-based Features 98
6.3.1.2 Character Embedding Features 99
6.3.2 Feature Selection 99
6.3.3 Multi-label Classification 100
6.3.3.1 Classifiers 100
6.3.4 Proposed model 100
6.3.5 Results and Discussion 103
6.3.5.1 Experiment Setup 103
6.3.5.2 Evaluation Metrics 103
6.3.5.3 Cross Validation 104
6.3.5.4 Ablation study of various feature sets with state-of-art classifiers 104
6.3.5.5 Multi-label classification: GA-BiSVM vs. BiSVM 106
6.3.5.6 Comparison of the proposed model with existing State-of-art-predictors on Benchmark dataset 107
6.3.5.7 Statistical Analysis 109
6.4 Protein Sub-Chloroplast Localization Prediction using Deep Neural Network 110
6.4.1 Feature Extraction 110
6.4.1.1 SkipXGram bi-gram (SXGbg) Technique 110
6.4.2 Multi-label Classification 111
6.4.2.1 Deep Neural Network 111

6.4.2.2 Hyper-parameter Optimization 113
6.4.3 Results and Discussion 114
6.4.3.1 Experiment Setup 114
6.4.3.2 Evaluation Metrics 114
6.4.3.3 Cross Validation 114
6.4.3.4 An Ablation Study on Evolutionary Profiles-based Features 114
6.4.3.5 Comparison with state-of-the-art models 116
6.4.3.6 Statistical Significance Analysis 119
6.5 Summary 120

7 Conclusion and Future Work **121**
7.1 Conclusion 121
7.2 Future Work 122

References **125**

List of Tables

2.1 Summary of State-of-the-art MSA Algorithms. 14

2.2 Summary of State-of-the-art PSSC Algorithms. 19

2.3 Summary of State-of-the-art PFR Algorithms. 23

2.4 Summary of State-of-the-art multi-label PSCL Algorithms. 27

3.1 Example of MPSA 32

3.2 Example of mutation count for a sample sequence pair 41

3.3 Pearson Correlation Coefficient Test on Aligned and Reference Scores . . 44

3.4 Average Q-Score comparison of the Proposed ProgSIO-MSA Model against the State-of-the-art Progressive Models on BAliBASE Dataset. 45

3.5 Average Q-Score Comparison of the Proposed ProgSIO-MSA Model against Progressive based Models on SABmark dataset. 45

3.6 The Q-Score Performance Comparison of the Proposed ProgSIO-MSA Model against Selected Genetic Algorithm based MSA Models on BAliBASE dataset. 47

3.7 Run Time Comparison of the Proposed ProgSIO-MSA Model against CLUSTAL Omega 47

3.8 Time Complexity Comparison of the Proposed ProgSIO-MSA Model against Iterative Model 48

3.9 Alignment Time(in milli secs) for ProgSIO-MSA and CLUSTAL Omega against the number of sequences of HomFam Test sets. 49

3.10 Wilcoxon Signed-Rank Test between the Proposed ProgSIO-MSA Model and other State-of-the-art Models on BAliBASE Dataset 51

4.1 Data Characteristics of Five Benchmark Datasets 54

4.2 Six Sets of Features from Proposed SXGbg Technique for a Sample Protein Sequence, S = KLMTPTRS 57

4.3 The Performance Comparison (in percentage) of Structural Class Prediction based on the CE Features Extracted from the PSIPRED against Hybrid CNN+RNN 61

4.4 The Performance Comparison of PSSC Prediction using SXGbg Features and Random Forest Classifier 62

4.5 The PSSC Prediction Evaluation (in %) of State-of-the-art Classifiers' using the Proposed Feature Modeling on the Three Benchmark Datasets 63

4.6 The Performance Comparison (in percentage) of the Proposed Model against State-of-the-art Methods for z277 Dataset 63

4.7 The Performance Comparison (in percentage) of the Proposed Model against State-of-the-art Methods for 25PDB Dataset 64

4.8 The Performance Comparison (in percentage) of the Proposed Model against State-of-the-art Methods for FC699 Dataset 64

4.9 Data Characteristics of Latest Large-Scale Low-similarity Dataset . 65

4.10 The Performance Comparison (in percentage) of various Embedding Architectures on the Benchmark Datasets 72

4.11 The Impact Analysis of the Proposed Feature Extraction Techniques on Benchmark Datasets using Ensemble of Classifiers 75

4.12 The Various Performance Metrics Result of the Proposed Model on Benchmark Datasets using 10-Fold Cross Validation 76

4.13 The Performance Comparison (in percentage) of the Proposed Model against State-of-the-art Methods for z277 Dataset 77

4.14 The Performance Comparison (in percentage) of the Proposed Model against State-of-the-art Methods for z498 Dataset 78

4.15 The Performance Comparison (in percentage) of the Proposed Model against State-of-the-art Methods for 25PDB Dataset 78

4.16 The Performance Comparison (in percentage) of the Proposed Model against State-of-the-art Methods for 1189 Dataset 79

4.17 The Performance Comparison (in percentage) of the Proposed Model against State-of-the-art Methods for FC699 Dataset 79

4.18 The Various Performance Metrics Result of the Proposed Model on Large-Scale Updated Dataset using 10-Fold Cross-Validation 80

4.19 The paired t-test among the Proposed Model and State-of-the-art Models from the Literature 81

5.1 The Information of all the Folds and the Frequency of the Protein Sequences in each Fold for the derived 25_SCOPe2.07_F36 Dataset . . 85

5.2 The Summary of Datasets that are used for Protein Fold Recognition 85

5.3 The Performance (in percentage) of Various Feature Sets On Derived Datasets 92

5.4 The Performance Comparison (in percentage) of the Proposed Conv-SXGbg-DeepFold Model against the State-of-the-art Models on DD, EDD and TG benchmark datasets. 94

5.5 The paired t-test among the Conv-SXGbg-DeepFold and two next-best models from the literature 95

6.1 Data Characteristics of Multi-Label Sub-Chloroplast Datasets 98

6.2 The Performance (Accuracy in percentage) Analysis of Evolutionary Feature sets using various State-of-art Classifiers. 105

6.3 The Performance (Accuracy in percentage) Analysis of Embedding Feature sets using various State-of-art Classifiers. 106

6.4 The Performance Comparison (in percentage) of GA-BiSVM and BiSVM on multi-label Benchmark Dataset using LOOCV 107

6.5 The Performance Comparison (in percentage) of the Proposed BiSVM Model with state-of-art predictors on Benchmark dataset using LOOCV 108

6.6 The Performance Comparison (in percentage) of the Proposed BiSVM Model with state-of-art predictors on Unknown Test Set, i.e., Novel dataset 108

6.7 The Paired t-test among the Proposed BiSVM and other three State-of-the-art Models on Benchmark dataset 109

6.8 The Performance (in percentage) of the Proposed Model on Various SXGbg Feature Sets Extracted from Evolutionary Profile of Benchmark Dataset. . 115

6.9 The Performance (in percentage) of the Proposed Model on Various Skipped-gram Feature Sets Extracted from Evolutionary Profile of Novel Dataset. . 116

6.10 The Performance Comparison (in percentage) of the Proposed Model against the State-of-the-art Models on Benchmark and Novel Datasets. 118

6.11 The Statistical Paired t-test between the Proposed Model and the State-ofthe-art Model on Overall Actual Accuracy 119

List of Figures

1.1 Cellular Activities from Genotype to Phenotype 2

2.1 A categorization of state-of-the-art alignment models updated with recent models. 10
2.2 Categorization of state-of-the-art MSA models based on the alignment approach. 13
2.3 Categorization of state-of-the-art PSSC models based on the feature source. 18
2.4 Categorization of state-of-the-art PFR models based on the feature source. 22
2.5 Categorization of state-of-the-art PSCL models based on the feature source. 26

3.1 Calculation of PRSDGP scores for two aligned profiles. 35
3.2 Progressive alignment using LBA and PRSDGP scoring strategies. 36
3.3 Steps to identify local optima ranges. 38
3.4 Workflow of single iterative optimization for local optima ranges. 40
3.5 Average Q-Score comparison on BAliBASE dataset 45
3.6 Scalability: Alignment time comparison of ProgSIO-MSA against CLUSTAL Omega for various number of sequences from HomFam test sets. Both axes are in logarithmic scales 50

4.1 The Proposed Model for the PSSC Prediction based on Local Features. 60
4.2 Framework of the proposed model constituting local and global features with ensemble of classifiers. 70
4.3 The Performance Variations of Embedding Architectures on Benchmark Datasets. 72
4.4 Overall Accuracy Variations on Feature Sets extracted from the Amino Acid Sequences for different Skip-gram (i.e., X) values. 73
4.5 Overall Accuracy Variations on Feature Sets extracted from the Secondary Structural Sequences for different Skip-gram (i.e., X) values. 74

5.1 The Proposed Framework consisting of Convolutional and Skip-Gram Feature Extraction techniques with Fully Connected Deep Neural Network for PFR. 89
5.2 The correlation of train and test accuracy of the proposed model for different epochs 93

6.1 Two-stage GA-BiSVM Classification Model for PSCL Prediction . 102

6.2 The Proposed Framework for PSCL consisting of Feature Extraction with Fully Connected Deep Neural Network. 112

6.3 The Performance Prediction of Various Skipped Gram Features on Benchmark Dataset 115

6.4 The Performance Prediction of Various Skipped Gram Features on Novel Dataset 116

6.5 The Performance Comparison of Evolutionary Profiles across Various Skipped Gram Features 117

6.6 The Performance Comparison of S5Gbg features of Evolutionary Profiles across Various Evaluation Metrics 117

6.7 The Performance Comparison (in percentage) of the Proposed Model against the State-of-the-art models 118

List of Abbreviations

3D	3-Dimensional
AAM	Amino Acid Molecule
AGP	Affine Gap Penalty
BiSVM	Binary Relevance with Support Vector Machine
BR	Binary Relevance
CATH	Class Architecture Topology and Homologous super-family
CBOW	Contiguous Bag-Of-Words
CCS	Conserved Column Score
CE	Character Embedding
Conv	Convolutional
DNA	Deoxyribonucleic Acid
fastText-CBOW	fastText Contiguous Bag-Of-Words
fastText-SG	fastText Skip-Gram
GA	Genetic Algorithm
GA-BiSVM	Genetic Algorithm based Binary Relevance with Support Vector Machine
GAGP	Global Affine Gap Penalty
GBM	Gradient Boosting Machine
GloVe	Global Vectors
GS	General Statistical
GSP	Global Sum-of-Pairs
HMM	hidden Markov model
k-NN	K-Nearest Neighbour
LAGP	Local Affine Gap Penalty
LBA	Look Back Ahead
LOOCV	Leave-one-out cross-validation
LOR	Local Optima Range
LR	Logistic Regression
LSP	Local Sum-of-Pairs
MC	Mutation Count
mGAGP	mean Global Affine Gap Penalty
mGSP	mean Global Sum-of-Pairs
MLP	Multi-Layer Perceptron
MPSA	Multiple Protein Sequence Alignment
MSA	Multiple Sequence Alignment
NJ	Neighbor-Joining
NGS	Next Generation Sequencing
NMR	Nuclear Magnetic Resonance
NP	Non-deterministic Polynomial

PFR	Protein Fold Recognition
ProgSIO-MSA	Progressive-based Single Iteration Optimization - Multiple Sequence Alignment
PRSDGP	Position-Residue Specific Dynamic Gap Penalty
PSCL	Protein Sub-Chloroplast Localization
PSI-BLAST	Position-Specific Iterative Basic Local Alignment Search Tool
PSIPRED	PSI-BLAST based secondary structure PREDiction
PSSC	Protein Secondary Structural Class
PSSM	Position-Specific Scoring Matrix
Q-Score	Quality-Score
RF	Random Forest
RNA	Ribonucleic Acid
SCOP	Structural Classification of Proteins
SCOPe	Structural Classification of Proteins - Extended
sdGAGP	standard deviation Global Affine Gap Penalty
sdGSP	standard deviation Global Sum-of-Pairs
sdLAGP	standard deviation Local Affine Gap Penalty
sdLSP	standard deviation Local Sum-of-Pairs
SG	Skip-Gram
SIO	Single Iterative Optimization
SP	Sum-of-Pair
SVM	Support Vector Machine
SXG-bg	SkipXGram bi-gram
TC	Total Column
TMCP	Total Mutation Count Pair-wise
UPGMA	Unweighted Pair Group Method with Arithmetic mean
W2V-CBOW	Word2Vec Contiguous Bag-Of-Words
W2V-SG	Word2Vec Skip-Gram

Chapter 1

Introduction

A cell is the basic unit of life that can replicate by itself, and cells are called the “building blocks of life”. Cells are categorized into two types: Prokaryotic cell which does not contain a nucleus, prokaryotes are single-celled organisms. A eukaryotic cell that contains a nucleus, eukaryotes are either single-celled or multi-cellular organisms Maton *et al.* (1997). In a eukaryotic cell, thread-like structures named chromosomes present inside the nucleus of animal and plant cells. Every chromosome consists of deoxyribonucleic acid (DNA) molecules, amino acid sequences (Proteins), and others.

Various metabolic activities are carried out by respective organelles that are in turn regulated by proteins Alberts *et al.* (1994). Proteins are biological macromolecules consisting of a combination of 20 unique amino acids in a sequence. Approximately one billion amino acid molecules reside in different subcellular locations of a cell Chou and Shen (2007).

Further, each organelle exhibits sub-compartments or sub-structures. E.g., the chloroplast location mainly consists of smaller substructures - thylakoid, envelope, stroma, plastoglobule, nucleoid. Thylakoid can be further divided into its lumen, membrane, and envelope into the inner and outer chloroplast membrane, including the intermembrane space. Each of these performs a specific function, e.g., the Thylakoid membrane carries out light reactions of photosynthesis. Sub-chloroplast localization of proteins provides insights into the roles of those proteins in the sophisticated photosynthesis process. Identifying proteins in these sub-compartments is even more difficult as this is one-level more microscopic in nature when compared to identifying proteins in compartments.

As shown in Figure 1.1, various cellular activities such as mutation, transcription, translation, and metabolites are involved in a cell. A mutation is a permanent alteration of one or more nucleotides in a gene sequence of a genome. Transcription is the primary step of gene expression, in which DNA segments called exomes are copied into ribonucleic acid (RNA) by the enzyme RNA polymerase. In translation, RNA is further translated to produce a specific amino acid chain/sequence or polypeptide. These polypeptides internally get folded due to various factors to become compact globular 3-Dimensional (3D) protein which performs respective functions in a cell. Metabolites regulate and perform chemical transformations within the cell.

All the above-mentioned activities play various, unique, and important roles in the overall functioning of a cell. From the above Figure 1.1, protein formation/generation is

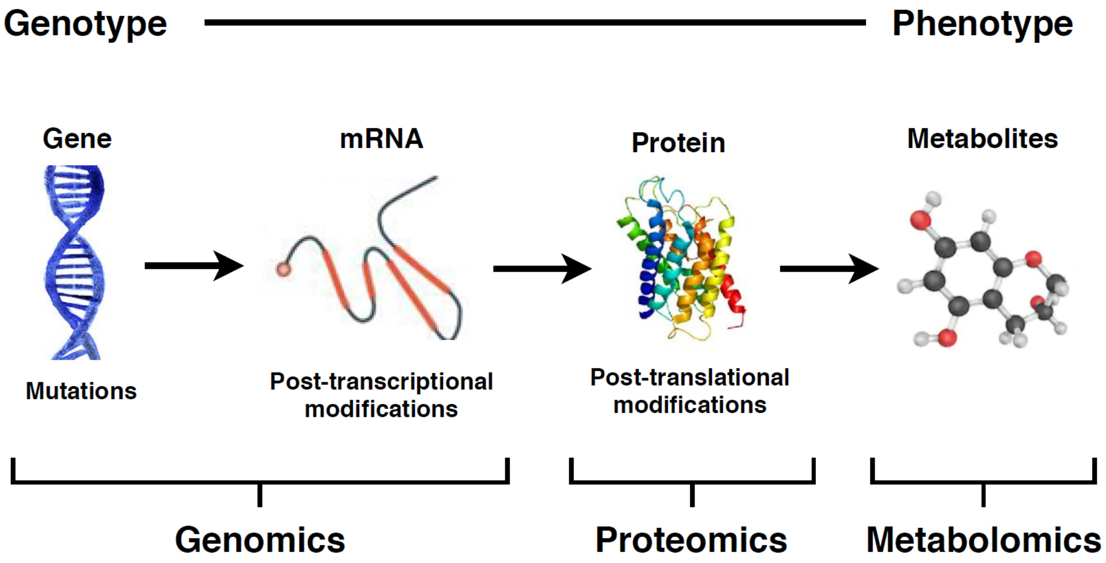


Figure 1.1. Cellular Activities from Genotype to Phenotype

one of the imperative aspects of the functioning life cycle of a cell. Hence, this research work mainly concentrates on analyzing amino acid sequences to solve protein structure and subcellular localization prediction problems.

Proteins are made up of a chain (linear) of amino acid molecules. Protein structure has been categorized into four levels, such as Primary Structure, Secondary Structure, Tertiary Structure, and Quaternary structure.

Primary structure: Primary structure of a protein refers to the linear sequence of amino acid molecules. The primary structure is also known as a polypeptide chain as the sequence of amino acid molecules are held together by peptide (covalent) bonds Lodish *et al.* (2000*a*).

Secondary structure: Secondary structure refers to highly regular local sub-structures on the actual polypeptide backbone chain. Two main types of secondary structure, i) the α-helix (H) and ii) the β-strand or β-sheets (E), were termed by Linus Pauling and co-workers in 1951 Pauling *et al.* (1951). These secondary structures are due to hydrogen bonds between the available hydrogen and oxygen atoms via the backbone peptide chain. Some subparts of the amino acid sequences do not contribute to any regular structure, termed as a random coil (C) Lodish *et al.* (2000*a*).

Tertiary structure: Tertiary structure refers to the 3D structure of monomeric and multimeric protein molecules. The α-helices and β-pleated-sheets are folded into a compact globular structure. This folding is carried away due to ionic interactions (van der walls Forces), hydrophobic, and hydrophilic interactions. However, the protein structure is said to be stable only when the protein becomes compact because of tertiary interactions Lodish *et al.* (2000*a*).

Quaternary structure: Quaternary structure is the 3D structure of multi subunits (two or more polypeptides) of protein and how the subunits fit together Lodish *et al.* (2000*a*).

Within a cell, proteins evolve from primary structure using various physico-chemical activities such as covalent bonding, hydrogen bonding, and ionic interactions. During this process. protein peptides move within a cell from one location to another location. Understanding the structure of proteins as well as its locations in a cell or within a cell organelle help in understanding their roles in the metabolic processes that are carried. Proteins with identical structure exhibit similar biological functions Chothia and Finkelstein (1990).

Many research works have been carried out to identify the protein structure and its subcellular localization. Biological methods as X-ray crystallography, nuclear magnetic resonance (NMR) spectroscopy is effective in identifying protein structure; whereas, cell fractionation, fluorescence microscopy, and electron microscopy techniques are accurate in protein subcellular localization prediction Chou and Shen (2007). The main advantage of biological methods is that they are accurate in identifying protein structures and its subcellular locations. The disadvantages of biological methods are that they are time-consuming and very expensive.

Protein data-banks such as Structural Classification of Proteins (SCOP), Class Architecture Topology and Homologous superfamily (CATH), and UniPort (Swiss-Prot) contains annotated protein information and they are publicly accessible. The SCOP is one of the largest publicly available protein data-banks in which proteins have been classified to determine the evolutionary relationship among proteins. The majority of proteins and their domains are manually curated of known structure in a hierarchy according to structural and evolutionary relationships.

Over the last two decades, the protein data in the protein data-banks have rapidly increased. It is worth noting that a total 31,474 number of protein domains were present in SCOP 1.55 version (released/updated before 2001-03-01) and, in SCOP 1.75 version (released/updated before 2009-02-23) the protein domains were increased to 110,800. As per the latest SCOP extended (SCOPe) 2.07 version (released/updated prior to 2019-10-03), a total of 303,552 protein domains are present Fox *et al.* (2013). From these statistics, we can say that the total number of protein domains in SCOP is increased approximately by 3 times per decade. Moreover, In 1986, Swiss-Prot Bairoch and Apweiler (2000) database had 3,939 protein sequences; whereas, now the number has grown to 560,537 according to the latest release by UniProtKB/Swiss-Prot UniProtKB/Swiss-Prot (2019). The number of protein entries is grown 142 times in the last 33 years.

The rapid increase of protein volumes is mainly due to the advancements in next-generation sequencing techniques. Due to the enormous increase of proteins in the protein data-banks, biological methods are not a feasible option to continue with as they are cost-intensive as well as time-consuming techniques. Due to this, various computational models were developed in the last couple of decades to identify the protein structure and its subcellular localization. However, the existing methods are not accurate. This study mainly concentrates on proposing effective computational models to solve the protein structure and its subcellular localization prediction.

1.1 Research Motivation

A prior accurate knowledge of protein structure and its subcellular location helps not only in finding answers to health issues but also helps in preventing many annotated issues in the field of medical science. Biological wet-lab experiments are time-consuming, challenging, and not cost-effective due to the need for high precision equipment. Biological experiments to identify protein structure and its subcellular localization for a large number of proteins is not a feasible approach. Moreover, sub-chloroplast localization increases the complexity of determining protein location as the location identification is one-level more microscopic in nature and the sub-compartments are much smaller in size compared to compartments. Computational methods are necessary to assist biologists in dealing with a large scale of proteins in a fast, efficient, easier, and cost-effective way. However, existing computational methods are less accurate in predicting the protein structure and its subcellular localization (detail discussion will be carried out in the next Chapter, i.e., Literature Survey, section 2). Therefore, this study mainly focuses on proposing effective computational models to solve the protein structure and its subcellular localization prediction.

1.2 Research Challenges

Multiple Sequence Alignment (MSA): is an important tool in the identification of evolutionary relations among the sequences and further contributes to the extraction of effective features. Existing MSA tools either effective in alignment quality or efficient with respect to time. An example of MSA is discussed in section 3.1. Aligning multiple sequences is not a trivial task because as the number of protein sequences increases, the alignment quality decrease, and also computational time increases. Over the last decade, many Genetic Algorithm (GA) based optimization methods such as GAPAM Naznin *et al.* (2012), RBT-GA Taheri and Zomaya (2009), SAGA Notredame and Higgins (1996), and others are adopted to increase the alignment quality; however, these methods run multiple iterations in refining the quality of the alignment; thus, the resources such as cpu cycles, time and space are also increases for every alignment.

Detail discussion on all the existing methods are carried out in the next Chapter, i.e., Literature Survey, at Section 2.1

Protein Secondary Structural Class Prediction: is an important and initial step in the identification of protein structure. Mainly there are four structural classes and each protein sequence can be categorized into any four of the classes. Existing methods are not generalized methods as they either work for high-similarity sequence datasets or low-similarity sequence datasets (detail discussion on existing methods are carried out in the next Chapter, at Section 2.2). Extracting important patterns that work effectively for all similarity datasets is the key to solve protein secondary strutural class prediction.

Protein Fold Recognition: is the next important step in the identification of protein structure. Every secondary structural protein is folded into the various structure and there are more than 30 different folds that can be possible. Existing methods are less effective (accuracy below 85% Lyons *et al.* (2016, 2015)) in extracting quality feature sets (detail discussion on existing methods are carried out in the next Chapter, at Section 2.3).

Protein Subcellular Localization Prediction: is one of the important prediction problems to identify the locality of a given protein. Every protein residing in a cell tends to move from one subcellular location to the other. In this study, locations of chloroplast proteins are identified and there are five different sub-chloroplast locations. Existing computational predictors fail to extract effective feature sets as well as, less effective in multi-label classification (detail discussion on existing methods are carried out in the next Chapter, at Section 2.4). The best predictor from the literature exhibits an accuracy of below 70% Wan *et al.* (2016*b*).

1.3 Major Contributions of the Thesis

The salient contributions of the research work are listed as follows:

- An effective progressive-based Multiple Protein Sequence Alignment (MPSA) method has been proposed. The importance of multiple sequence alignment is that a vital and consistent homologous pattern of proteins can be captured and these patterns will further help in solving protein structure and its subcellular locations. The proposed MPSA method involves the following:
    - An effective scoring system that contains two novel scoring strategies named, Look Back Ahead (LBA) and Position-Residue Specific Dynamic Gap Penalty (PRSDGP) have been proposed to improve the alignment quality.
    - An effective and novel optimization framework named Single Iterative Optimization (SIO) has been proposed to refine the alignment quality.

- An enhanced Protein Secondary Structural Class (PSSC) prediction model has been proposed which involves the following:
    - An effective feature modeling to capture local and global amino acid interactions. Local interactions are captured from primary and secondary structural sequences using two novel techniques such as Character Embeddings (CE) and SkipXGram bi-gram (SXG-bg); Whereas, global interactions are captured via frequency approach.
    - Various conventional state-of-the-art classifiers are explored for the prediction.
    - An effective ensemble of classifiers is proposed to predict PSSC accurately.
- An enhanced Protein Fold Recognition (PFR) model for low similarity datasets has been proposed which involves:
    - An effective feature extraction approach that includes extracting local features from evolutionary-based profiles using SkipXGram bi-gram (SXGbg) technique and global features using convolutional operations.
    - An effective deep learning architecture to predict PFR accurately.
- An effective multi-label Protein Sub-Chloroplast Localization (PSCL) prediction model has been proposed which involves:
    - An effective feature extraction approach that includes extracting local features from evolutionary-based profiles using SkipXGram bi-gram (SXGbg) technique.
    - An effective multi-label deep learning architecture to predict PSCL accurately.

1.4 Organization of the Thesis

Chapter 2 describes a detailed literature survey on MPSA methods, PSSC prediction models, PFR models, and PSCL prediction models and followed by the problem statement and research objectives.

Chapter 3 describes the proposed methodology of progressive-based MPSA in detail. In this chapter, two effective scoring strategies are discussed followed by SIO optimization framework to refine the alignment. Further, a total time complexity of the proposed MPSA model is discussed along with the performance analysis.

Chapter 4 elaborates the proposed PSSC prediction model in detail. In this chapter three feature extraction techniques followed by an effective ensemble of classifiers are discussed. Further, the performance analysis of the proposed model is assessed on benchmark and derived dataset and evaluated against state-of-the-art models.

Chapter 5 discusses in detail the proposed PFR prediction model. In this chapter two feature extraction techniques followed by an effective deep learning framework are discussed. Further, the performance analysis of the proposed model is assessed on benchmark and derived datasets and evaluated against state-of-the-art models.

Chapter 6 proposes two multi-label PSCL prediction models. In this chapter, a preliminary study is discussed that includes PSSM-based bi-gram features followed by a Binary Relevance framework containing SVM as a base classifier to predict the multi-label locations of PSCL. Further, the SkipXGram bi-gram feature extraction technique followed by a deep learning framework is discussed. The performance analysis of the proposed model is assessed on benchmark datasets and evaluated against state-of-the-art models.

Conclusions and possible future directions are highlighted in chapter 7.

1.5 Summary

This chapter highlights cells, followed by various metabolic activities that are carried-out in cell survival. Various kinds of protein structures, protein subcellular localization, and their importance have been highlighted. This chapter lists various biological techniques to identify protein structure and its subcellular localization and followed by its limitations due to the high volumes of proteins to be analyzed. The research motivation and the major contributions of the overall study are listed out in this chapter. The chapter also highlights the organization of the thesis.

In the next chapter, some of the important existing work in the area of MSA, PSSC prediction, PFR prediction, and PSCL prediction has been reviewed.

Chapter 2

Literature Survey

An extensive literature survey was carried out in the field of protein structure and its subcellular localization which includes areas like, multiple protein sequence alignment, protein secondary structure class prediction, protein fold recognition, and subcellular localization prediction.

2.1 Multiple Protein Sequence Alignment

Multiple Sequence Alignment (MSA) is one of the fundamental tools in molecular biology which aligns more than two biological sequences simultaneously. The alignment of amino acid sequences is termed as Multiple Protein Sequence Alignment (MPSA). Based on the similarities of the amino acid sequences, the core regions are identified along with possible alterations over the years. Sequence alignment helps to identify homology (existence of shared ancestry) and divergence patterns between new and existing sequences. These patterns in-turn helps in various biological activities such as i) in the reconstruction of phylogenetic trees, ii) in predicting the secondary structural class information, iii) in recognition of protein folds, iv) in the identification of protein tertiary structures, and iv) also to predict subcellular localization of given amino acid sequence Notredame (2002).

The various match, mismatch, and indel ("-") events help in organizing related sequences that are possibly deviated via evolution. Aligning two sequences termed as pair-wise alignment. The goal is to find similar or closely related regions in pair-wise alignment Naznin *et al.* (2012). For short lengths and a smaller number of sequences, possibly the alignment can be achieved manually. However, for the number of sequences more than 8 and above, efficient methods are essential for alignments Thompson *et al.* (1994).

The existing MSA models are categorized mainly into three approaches: (i) classical approach, (ii) progressive approach, and (iii) iterative approach. Categorization of various progressive and iterative-based state-of-the-art models as illustrated by Thompson *et al.* (1999) and is further updated with the recent methods (highlighted in blue color) is as shown in Figure 2.1.

(i) Classical Approach: adopts a dynamic programming approach to align a sequence pair by computing every probable option to obtain optimal alignment. Needleman and Wunsch (1970) proposed global alignment technique and Smith and Waterman (1981) proposed local alignment technique. Both global and local pairwise alignment

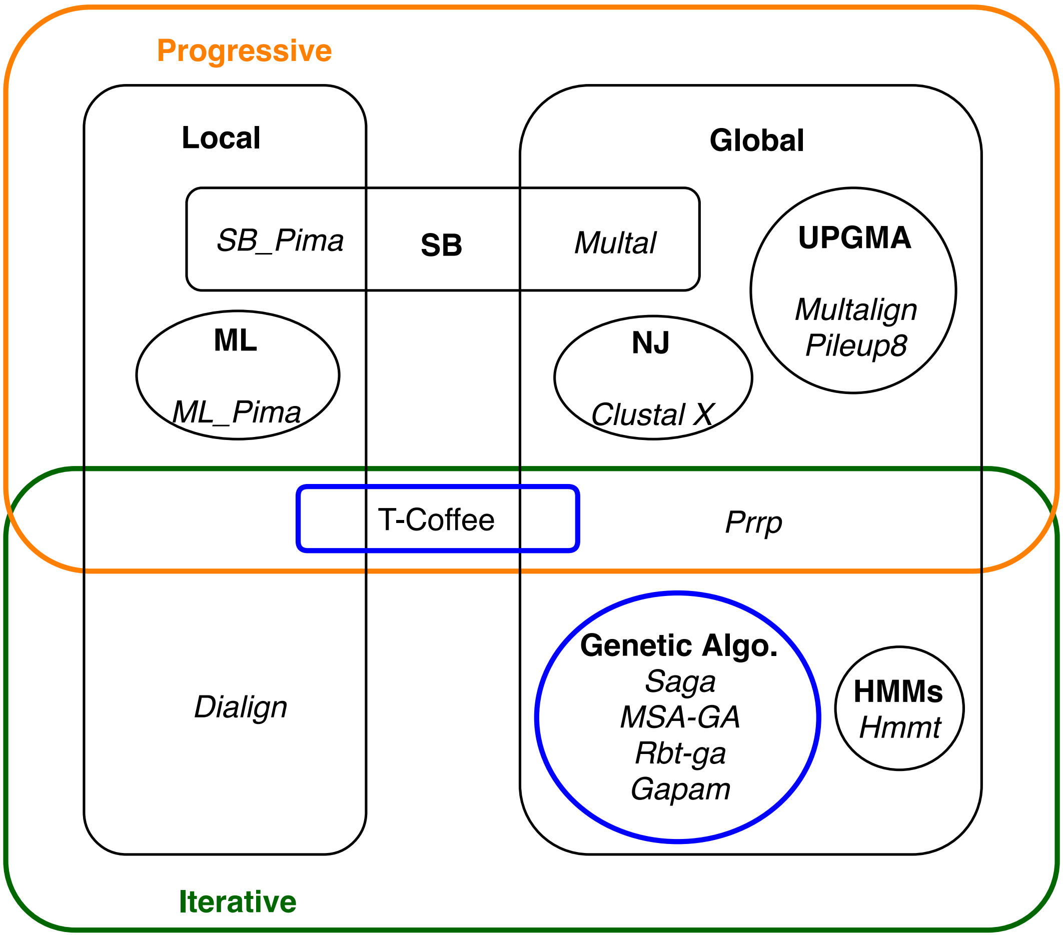


Figure 2.1. A categorization of state-of-the-art alignment models updated with recent models.

techniques are the standard examples of the classical approach.

The global alignment (Needleman-Wunsch) method aligns a sequence pair over their entire length Needleman and Wunsch (1970), whereas, the local alignment (Smith-Waterman) method identifies similar regions within a sub sequence to align core blocks Smith and Waterman (1981).

Advantages: Classical approach aligns the biological sequences in an optimum way.

Time Complexity: The time complexity to align a sequence pair optimally using classical approach is O(L^2), where L is the maximum length of the given sequence pair.

Limitations: As the number of sequences (n) increases, the time complexity to find the optimal alignment becomes O($n.2^n.L^n$) Waterman *et al.* (1976). Moreover, optimizing MSA problem is considered as non-deterministic polynomial time (NP)-complete problem Wang and Jiang (1994).

(ii) Progressive Approach: To overcome the limitation of the classical approach, the authors, Feng and Doolittle (1987) introduced to progressive alignment approach. A progressive-based model selects and aligns the best sequence pair based on similarity or distance measure among the sequences. The next best pair is selected and aligned

progressively until all the sequences are aligned.

A progressive model to obtain accurate alignment depends on two factors: (i) a scoring system to find the optimal arrangement among the sequences based on a score; and (ii) the order in which the sequence pair is selected for alignment. For the first factor, a majority of the developed scoring systems are variants of the sum-of-pair (SP) scoring function Wang and Jiang (1994). For the second factor, Neighbor-Joining (NJ) Saitou and Nei (1987) and Unweighted Pair Group Method with Arithmetic mean (UPGMA) Sneath and Sokal (1973) are widely accepted techniques.

Various progressive-based models have been developed such as: PIMA Smith and Smmith (1992) aligns only the conserved blocks using local alignment approach. To decide the order in which sequences are selected for alignment, it uses maximum linkage (ML_PIMA) or sequential branching (SB_PIMA). Align-M Van Walle *et al.* (2004*a*) aligns in three steps: first, it performs with high-scoring local alignment; next, one or more pair-wise alignments performed by dynamic programming using mutation matrix; and finally, only consistent alignment parts are considered for final alignment. Both CLUSTAL X Thompson *et al.* (1997) and CLUSTAL W Thompson *et al.* (1994) adopts NJ technique to build tree to solve MSA. CLUSTAL X Thompson *et al.* (1997) provides a graphical interface for CLUSTAL W Thompson *et al.* (1994). Although it has the sophisticated scoring system, it suffers from local optima trap Notredame *et al.* (2000). MULTALIGN Barton and Sternberg (1987) and PILEUP Devereux *et al.* (1984) follow global alignment approach, in which both solves MSA with the help of guide tree which is constructed using UPGMA Sneath and Sokal (1973).

Advantages: Progressive models align the biological sequences in polynomial time.

Time Complexity: Time complexity to align sequences using progressive approach with UPGMA Sneath and Sokal (1973) is O(n^3+$n.L^2$) Edgar (2004*a*). Where, n is the number of sequences and L is the maximum length of a sequence.

Limitations: Progressive-based alignment follows heuristic approach and tends to exhibit – ‘once a gap, always a gap’ Feng and Doolittle (1987). This limitation is due to the greediness of the progressive approach which leads to local optima.

(iii) Iterative Approach: To overcome the local optima limitation, many iterative-based optimization models have been developed. The iterative approach adopts either progressive or stochastic techniques, in which initial alignment is constructed and re-aligned iteratively until the alignment converges to optimum alignment or runs for maximum iterations.

Various progressive-based iterative models have been developed such as MAFFT Katoh and Standley (2013) incorporates fast Fourier transform technique to perform progressive based (FFT-NS-2) alignment and iterative based (FFT-NS-i) alignment. MUSCLE Edgar (2004*b*) performs MSA in three stages - initially draft progressive, followed by improved progressive, and finally refinement. KAlign2 Lassmann *et al.* (2008) is an improved version in which consistency check is avoided by using external features. CLUSTAL Omega Sievers *et al.* (2011) uses external profile alignment which is based on a hidden Markov model to solve MSA. T-Coffee Notredame *et al.* (2000) falls under consistency-based iterative category in which it adopts both global and local alignment techniques to align and refines the final alignment using a library. SAGA Notredame and Higgins (1996), RBT-GA Taheri and Zomaya (2009), MSA-GA Gondro and Kinghorn (2007), GAPAM Naznin *et al.* (2012) and MSAGMOGA Kaya *et al.* (2014) are examples of genetic algorithm (GA) based iterative approach. SAGA Notredame and Higgins (1996) optimizes the solution using 22 different operators repeatedly, due to which the method suffers from high order time complexity Shyu *et al.* (2004). RBT-GA Taheri and Zomaya (2009) is a heuristic-based approach to align sequences using a dynamic programming table. In MSA-GA Taheri and Zomaya (2009), the initial population is generated using Thompson *et al.* (1994) and pair-wise alignment. GAPAM Naznin *et al.* (2012) performs MSA using guide trees, where the initial population is generated with two approaches: first, random guide trees; and second, shuffling sequences among trees. MSAGMOGA Kaya *et al.* (2014) optimizes the three objectives: similarity maximization, affine gap penalty minimization, and support maximization to improve overall alignment quality.

Advantages : The probability of an alignment suffering from local optima is very minimal.

Time Complexity : Time complexity to construct initial alignment is O(n^3+$n.L^2$) with a maximum number of iterations, denoted as K, to optimize. The overall time complexity of the iterative approach is O($K.n^3$+$K.n.L^2$).

Limitations : Iterative-based models might converge to optimal alignment with the trade-off of higher time complexity.

Scoring System :

Along with the alignment approaches, the scoring system also plays a crucial role in the alignment quality. Many efforts are made to improve the robustness of the scoring system, of which Hierarchical Expected matching Probability (HEP) Nguyen and Pan (2011) is one among the promising approaches.

Sum-of-pairs (SP) is also a promising approach, in which most models were adopted due to its simplicity. CLUSTAL W Thompson *et al.* (1994), T-Coffee Notredame *et al.* (2000) and MAFFT Katoh and Standley (2013) are models which make use of SP scoring strategy with variations in substitution or mutation matrices like PAM Dayhoff *et al.* (1978), BLOSUM Henikoff and Henikoff (1992) and GONNET Gonnet *et al.* (1992). The scores of these mutation matrices are derived from various residue factors such as physicochemical properties, frequency, entropy and mutation information Valdar (2001).

One of the most commonly used approaches to penalize indels is the Affine Gap Penalty (AGP). In MUSCLE Edgar (2004*b*), gap penalty depends on two concepts, i.e., gap open and gap extend, which are calculated based on an input value and a tune-able parameter, 'H'. In CLUSTAL W Thompson *et al.* (1994), besides position-specific gap penalty, additional factors like sequence length, sequence similarity, and average residue mismatch are considered to calculate the gap penalty. In MAFFT Katoh and Standley (2013) and H4MSA Rubio-Largo *et al.* (2016), the gap penalty is a static value provided as part of alignment input, and identifying the fixed value is not a trivial task.

The overall important works of MSA are summarized in Table 2.1 and the categorization of these works based on the alignment approach are shown in Figure 2.2.

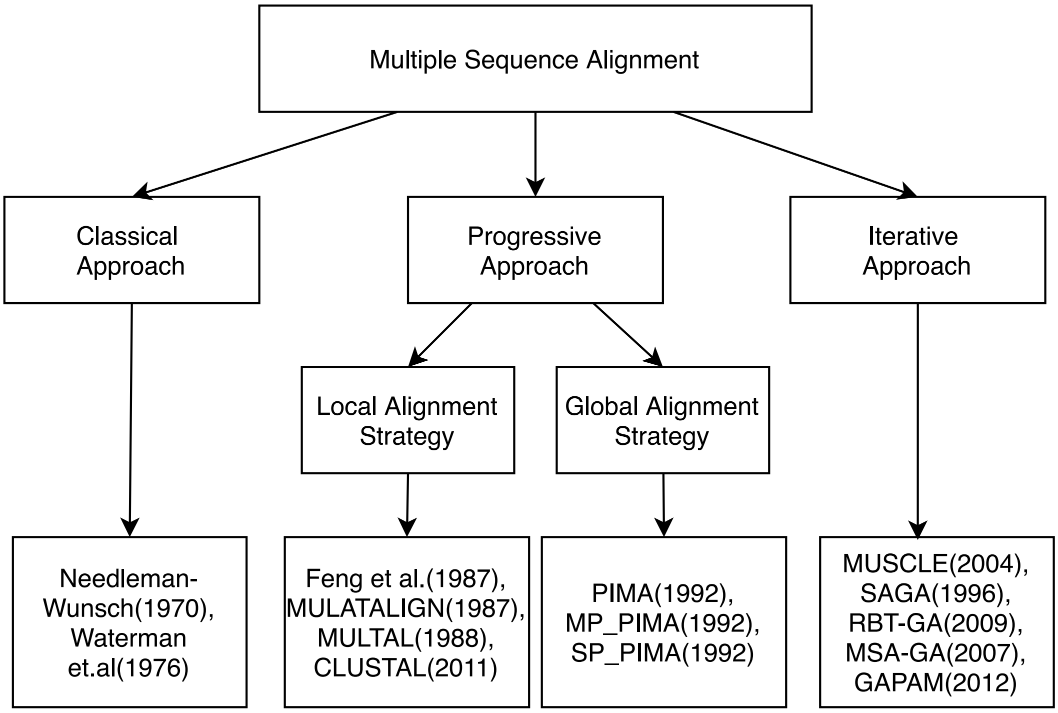

Multiple Sequence Alignment
Classical Approach
Progressive Approach
Iterative Approach
Local Alignment Strategy
Global Alignment Strategy
Needleman-Wunsch(1970), Waterman et.al(1976)
Feng et al.(1987), MULATALIGN(1987), MULTAL(1988), CLUSTAL(2011)
PIMA(1992), MP_PIMA(1992), SP_PIMA(1992)
MUSCLE(2004), SAGA(1996), RBT-GA(2009), MSA-GA(2007), GAPAM(2012)

Figure 2.2. Categorization of state-of-the-art MSA models based on the alignment approach.

..

Table 2.1. Summary of State-of-the-art MSA Algorithms.

Approach	**Methods**	**Remarks**	**Limitations**
Classical	Needleman and Wunsch (1970), Waterman *et al.* (1976)	Accurately aligns all the given sequences.	Time complexity increases exponentially as the number of sequences are increased.
Progressive Global Alignment	Feng and Doolittle (1987), MULTALIGN Barton and Sternberg (1987), MULTAL Taylor (1988), CLUSTAL Sievers *et al.* (2011)	Fast, Deterministic, and suited for closely related species.	Doest work well for distantly related species. Suffers from local optima.
Progressive Local Alignment	PIMA, MP_PIMA, SP_PIMA Smith and Smmith (1992)	Fast, Deterministic, and suited for dataset which has highly correlated core blocks over the sequences.	Not reliable outside core residue blocks. Suffers from local optima.
Iterative	MUSCLE Edgar (2004*b*), SAGA Notredame and Higgins (1996), RBT-GA Taheri and Zomaya (2009), MSA-GA Gondro and Kinghorn (2007), GAPAM Naznin *et al.* (2012)	Follows heuristics approach, avoids local optima trap, feasible results	Iterates for maximum number to converge at global optima, which is time consuming.

2.2 Protein Secondary Structural Class

Prior knowledge of protein secondary structural class (PSSC) information helps in discovering its structure and functions Chou (2000).

In 1976, the authors, Levitt and Chothia (1976) proposed the concept of protein secondary structural classes based on the visual examination of polypeptide chain topologies, which are categorized into four structural classes from a dataset of 31 globular proteins: all-α , all-β , α/β, and α+β. While the first two classes comprise secondary structures dominated by α–helices and β–strands, respectively, the other two classes consist of both α–helix and β–strand secondary structures with interspersed in α/β class structures and segregated in α+β class structures. Structural Classification of Proteins (SCOP) Andreeva *et al.* (2004) and Class Architecture Topology and Homologous superfamily (CATH) Cuff *et al.* (2009) are two protein structure databases that provide hierarchical structural classifications of proteins. The classification of proteins to respective structural classes in the SCOP database is manually validated, while in the CATH database it is validated via both automated as well as manual procedures.

New protein structures discovered by diverse scientific communities have been submitted to protein databases. According to the latest extended version of the SCOPe 2.07 database[1], the proteins are mainly categorized into seven classes, namely, (1) All–α, (2) All–β, (3) α/β, (4) α+β, (5) Multi-domain proteins, (6) Membrane and cell surface proteins and (7) Small proteins. Over the years, it was observed that 90% of these protein entries consistently belong only to the first four structural classes Murzin *et al.* (1995); Andreeva *et al.* (2004), Andreeva *et al.* (2007); Fox *et al.* (2013). Therefore, this study mainly concentrates on predicting the first four structural classes.

Identification of protein structural class is one of the important activities of protein sequence analysis for mainly two reasons: (i) Prior knowledge of the structural class information of protein sequences enhances the prediction accuracy of several activities of sequence analysis such as DNA-binding sites Kuznetsov *et al.* (2006), protein secondary structure Rahal and Walz (2018), protein folds Aram and Charkari (2015); Raicar *et al.* (2016); Ibrahim and Abadeh (2017, 2018), protein folding rates Gromiha (2005), tertiary structure prediction Carlacci *et al.* (1991); all these activities have potential applications in further analysis of protein functions and drug discovery Chou *et al.* (2006). (ii) Newly discovered protein sequences from the various scientific communities are consistently increasing due to the rapid advancement of sequencing technology. Hence, to annotate the structural class information for newly discovered protein sequences, there is an imminent need for automated, accurate, and generalized

---

[1] `http://scop.berkeley.edu/statistics/ver=2.07`

structural class prediction models that works for all categories of sequence similarity proteins.

Earlier investigations on the identification of PSSC were carried out by biological experimental methods Provencher and Gloeckner (1981). However, these methods are time-consuming and cost-intensive. To overcome the limitations of biological methods, several computational methods have been proposed Klein and Delisi (1986); Liu *et al.* (2010*b*); Yu *et al.* (2013); Dehzangi *et al.* (2013*b*); Kavianpour and Vasighi (2017).

From the last two decades, various computational efforts have been made to figure out the protein secondary structure class prediction Klein and Delisi (1986); Liu *et al.* (2010*b*); Yu *et al.* (2013); Dehzangi *et al.* (2013*b*); Kavianpour and Vasighi (2017). The majority of these are machine learning methods, considering prediction secondary structure as a classification problem with four classes. The PSSC prediction is categorized under a multiclass classification problem, which involves two major activities: (i) Feature modeling and (ii) Classification.

In feature modeling activity, the given sequences are transformed into fixed-length feature vectors and relevant features are identified to predict the PSSC accurately. In literature, state-of-the-art models extract features mainly from amino acid sequences, structural sequences, and evolutionary information.

Sequence-based features are primarily extracted from information such as physicochemical properties of protein residues Zhang *et al.* (2008); Rahal and Walz (2018); Contreras-Torres (2018), amino acid composition (AAC) and their distribution Klein and Delisi (1986), pseudo amino acid composition (PseAAC) Chen *et al.* (2006); Zhang *et al.* (2008); Li *et al.* (2009); Sahu and Panda (2010), and averaged chemical shifts Zhu *et al.* (2019). The advantage of sequence-based features is that they exhibit significant discriminating information for high similarity datasets. In contrast, sequence-based features fail to discriminate classes correctly for twilight zone (low similarity) datasets Kurgan and Homaeian (2005); Kurgan and Chen (2007); Mizianty and Kurgan (2009).

Structure-based features are extracted from secondary structural sequences. The secondary structural sequence can be generated by mapping every amino acid residue from the protein sequence to one of the secondary structure elements such as α-Helix (H), β-Sheet (E), or Coil (C) McGuffin *et al.* (2000). Liu et al. Liu and Jia (2010) focused on designing features from structural sequences. Kong et al. Kong *et al.* (2014) extracted features to characterize general contents and spatial arrangements of the secondary structural sequences. PSSC prediction methods Liu and Jia (2010); Kong *et al.* (2014) using secondary structural sequences reported better prediction accuracy. However, these methods were not able to explore and extract highly discriminating features.

Evolution-based features are extracted from sequence profiles such as position-specific scoring matrix (PSSM) which are generated using PSI-BLASTAltschul *et al.* (1997). To address PSSC prediction, various techniques are applied to evolutionary information; Zang et al. Zhang *et al.* (2012) extracted a large vector space and reduced it using the principal component analysis approach. Xia et al. Xia *et al.* (2012) work focus on transforming evolutionary features using the linear regression technique. Liu et al. Liu *et al.* (2012) adopt auto-covariance transformation technique on PSSM. Dehzangi et al. Dehzangi *et al.* (2013*b*) extracted features from both physicochemical properties and evolutionary information using overlapped segmented distribution and autocorrelation techniques. Zang et al. Zhang *et al.* (2014) extracted features based on evolutionary differences. Ding et al. Ding *et al.* (2014) extracted long-range and linear correlation features from evolutionary information. Qin et al. Qin *et al.* (2015) generated a fixed-length feature vector by the linear predictive coding approach.

Other approaches in solving the PSSC problem include: Liu et al. Liu *et al.* (2010*b*) addressed the PSSC problem using a distance measure instead of extracting discriminating features. Yu et al. Yu *et al.* (2013) parallelly extracted features from multiple views and fused them to form a complex feature space. Kavianpour et al. Kavianpour and Vasighi (2017) transformed amino acid residues to binary codes based on the hydrophobicity index and then generates cellular automata images to extract features using image textural properties.

The latter activity of PSSC prediction i.e., for the classification, various state-of-the-art classifiers such as Bayesian classifier Wang and Yuan (2000), Logistic Regression Kurgan and Homaeian (2006); Kurgan and Chen (2007), Artificial Neural Network Cai and Zhou (2000); Sahu and Panda (2010); Ningbo and Hua (2017), ensemble classifiers Kedarisetti *et al.* (2006); Dehzangi *et al.* (2013*b*), and Support Vector Machine Kurgan and Chen (2007); Zhang *et al.* (2012); Liu *et al.* (2012); Zhang *et al.* (2014); Ding *et al.* (2014); Qin *et al.* (2015) have been developed for PSSC prediction.

In the literature, the supervised learning techniques such as SVM and other ensemble classifiers have been widely adopted to solve the PSSC problem.

The overall important works of PSSC prediction are summarized in Table 2.2 and the categorization of these works based on the feature sources are shown in Figure 2.3.

.

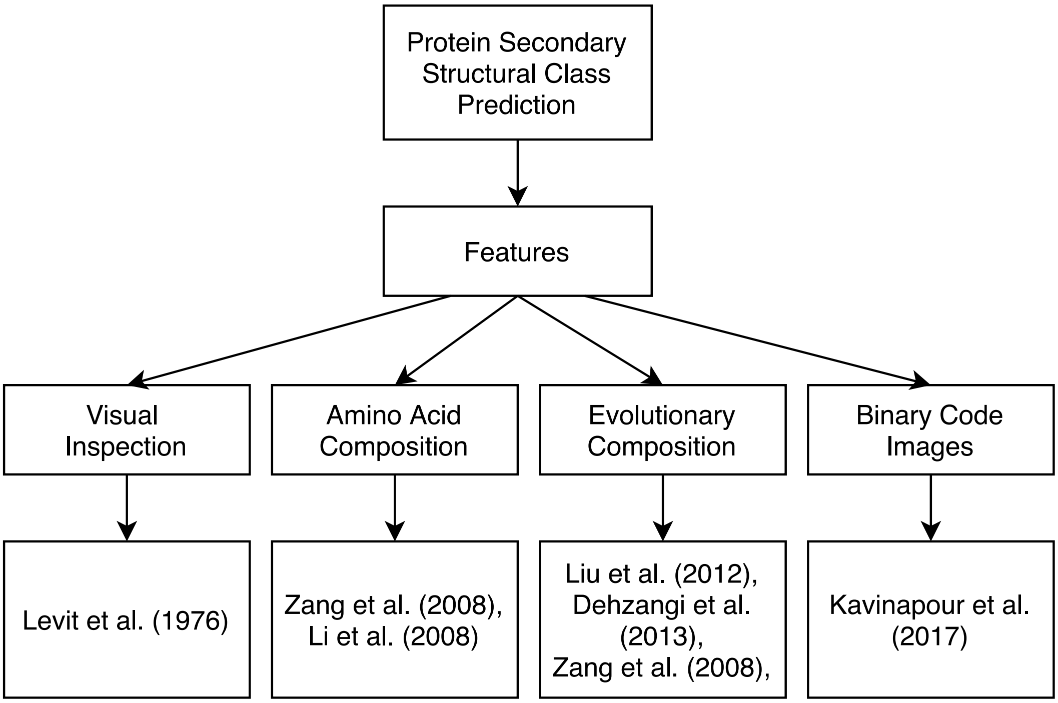

Protein Secondary Structural Class Prediction
Features
Visual Inspection
Amino Acid Composition
Evolutionary Composition
Binary Code Images
Levit et al. (1976)
Zang et al. (2008), Li et al. (2008)
Liu et al. (2012), Dehzangi et al. (2013), Zang et al. (2008),
Kavinapour et al. (2017)

Figure 2.3. Categorization of state-of-the-art PSSC models based on the feature source.

Table 2.2. Summary of State-of-the-art PSSC Algorithms.

Methods	**Approach**		**Remarks**
	Features	**Classification**	
Michael Levitt Levitt and Chothia (1976)	Experimental Approach: Visual inspection of the topologies of polypeptide chains	Manually Classified	Precise and Accurate. Time Consuming & Cost intensive.
Zhang Zhang *et al.* (2008), Li et al. Li *et al.* (2009)	Physicochemical-Based, Amino Acid and Pseudo Amino Acid Composition	fuzzy K nearest neighbors, support vector machine	Doesnt capture local information features such as evolutionary and structural based features.
Liu et al. Liu *et al.* (2012), Dehzangi et al. Dehzangi *et al.* (2013*b*), Zang et al. Zhang *et al.* (2014)	Auto-covariance transformation on PSSM, Physicochemical + PSSM, Evolutionary differences	support vector machine, ensemble classifier, support vector machine	Many works explored PSSM based features and obtained satisfactory results.
Liu et al. (2010) Liu and Jia (2010)	Features are extracted from secondary structural sequences	support vector machine	Less explored approach
Kavianpour et al. Kavianpour and Vasighi (2017)	Transformed amino acid residues to binary codes. Built cellular automata images. Features were extracted from texture properties.	support vector machine	Feature extraction process is time consuming. Works for high sequence similarity datasets only.

2.3 Protein Fold Recognition

The protein fold recognition (PFR) is one step closer in identifying protein tertiary structure. The PFR refers to an assignment of a query protein sequence to one of the structural folds (from a limited number of folds) that exhibits a similar tertiary structure. Proteins with identical structure exhibit similar biological functions Chothia and Finkelstein (1990).

Biological wet-lab methods such as nuclear magnetic resonance (NMR) and X-ray crystallography are proven to be effective in the identification of protein tertiary structure with a trade-off of higher expenses and time Berardi *et al.* (2011).

Moreover, with recent improvements in large-scale sequencing technologies, a massive number of protein sequences have been deposited in protein data banks. In 1986, Swiss-Prot Bairoch and Apweiler (2000) database had 3,939 protein sequences; whereas, now the number has grown to 560,537 according to the latest release by UniProtKB/Swiss-Prot UniProtKB/Swiss-Prot (2019). The number of protein entries is grown 142 times in the last 33 years. The experimental methods become an infeasible solution to handle the massive growth of protein sequences; thus, effective computational approaches are highly desirable.

Over the last two decades, numerous computational approaches have been developed by various research communities and among these approaches, machine learning models have been achieved promising results. In machine learning terminology, the PFR falls under the category of supervised learning with a multiclass classification problem. The performance of the PFR mainly depends on feature extraction and classification techniques.

In literature, a wide range of features are utilized for PFR and these features are broadly categorized into four groups, namely: (i) Sequence-based features Chothia and Finkelstein (1990); Shen and Chou (2006); Lin *et al.* (2007); Ying *et al.* (2009); Yang *et al.* (2011); Kavousi *et al.* (2011): where features are extracted from amino acid and/or pseudo amino acid compositions, (ii) Physicochemical-based features Ding and Dubchak (2001); Dehzangi and Phon-Amnuaisuk (2011); Sharma *et al.* (2013*b*); Aram and Charkari (2015): where features are extracted from the physicochemical properties of amino acids, (iii) Structural-based features Chen and Kurgan (2007); Shen and Chou (2009); Chen *et al.* (2011); Paliwal *et al.* (2014*a*): where the features are extracted on structural information of a protein sequence, and (iv) Evolutionary-based features Kavousi *et al.* (2011); Chen and Kurgan (2007); Dong *et al.* (2009); Yang and Chen (2011); Sharma *et al.* (2013*a*); Paliwal *et al.* (2014*b*); Lyons *et al.* (2015): where the features are derived from evolutionary profiles.

Ding and Dubchak Ding and Dubchak (2001) considered amino acid composition as well as physicochemical attributes with structural information for feature extraction, which achieved 56% accuracy in predicting protein fold. Earlier studies were mainly focused on sequence-based, and physicochemical-based features, and the performance of PFR was limited to below 70% accuracy for low similarity (i.e., $<$ 40% similarity) benchmark datasets. Later studies have explored structural-based and evolutionary-based features to enhance the PFR accuracy above 70% for low similarity benchmark datasets. Recent studies Lyons *et al.* (2016, 2015) using evolutionary-based hidden Markov model (HMM) profile features further enhanced the PFR results and reported just above 80% accuracy for low similarity benchmark datasets. It is worth noting that the HMM profiles have been playing a key role in the performance enhancement of various protein prediction challenges Kumar *et al.* (2020).

Most features are extracted by exploring various techniques such as, auto covariance Dehzangi *et al.* (2013*a*, 2014); Ibrahim and Abadeh (2017), autocross-covariance Dong *et al.* (2009); Yan *et al.* (2017), auto-correlation Shen and Chou (2006); Ibrahim and Abadeh (2017), mono-gram Sharma *et al.* (2013*a*); Lyons *et al.* (2015), bi-gram Sharma *et al.* (2013*a*); Lyons *et al.* (2015), tri-gram Paliwal *et al.* (2014*b*); Lyons *et al.* (2015), pairwise frequency information Ghanty and Pal (2009), k-amino acid pairs Paliwal *et al.* (2014*a*), distance between evolutionary profiles Lyons *et al.* (2014, 2016), and fusion of different features Ding and Dubchak (2001); Shen *et al.* (2009); Ghanty and Pal (2009); Yan *et al.* (2017); Ibrahim and Abadeh (2017). The best PFR accuracies that are reported on benchmark datasets are using tri-gram features Lyons *et al.* (2015).

For classification, various machine learning techniques have been developed such as linear discriminant analysis Ibrahim and Abadeh (2017, 2018), Bayesian classifier Chinnasamy *et al.* (2005), support vector machine (SVM) Ding and Dubchak (2001); Dong *et al.* (2009); Yang and Chen (2011); Sharma *et al.* (2013*a*); Paliwal *et al.* (2014*a*); Lyons *et al.* (2014); Dehzangi *et al.* (2014); Lyons *et al.* (2015), k-nearest neighbor Shen and Chou (2006), Artificial neural network Cai and Zhou (2000), Extreme learning machine Ibrahim and Abadeh (2017, 2018), hidden Markov model Bouchaffra and Tan (2006); Deschavanne and Tufféry (2009), and ensemble of different classifiers Shen and Chou (2006); Ghanty and Pal (2009); Dehzangi *et al.* (2010); Aram and Charkari (2015). The SVM and ensemble of SVM with other classifiers have reported promising results Dehzangi *et al.* (2010); Lyons *et al.* (2015).

In literature, various combinations of feature extraction and classification techniques have been proposed to solve PFR effectively. However, the PFR results for low similarity benchmark datasets are still below 85%.

The overall important works of PFR are summarized in Table 2.3 and the categorization of these works based on the feature sources are shown in Figure 2.4.

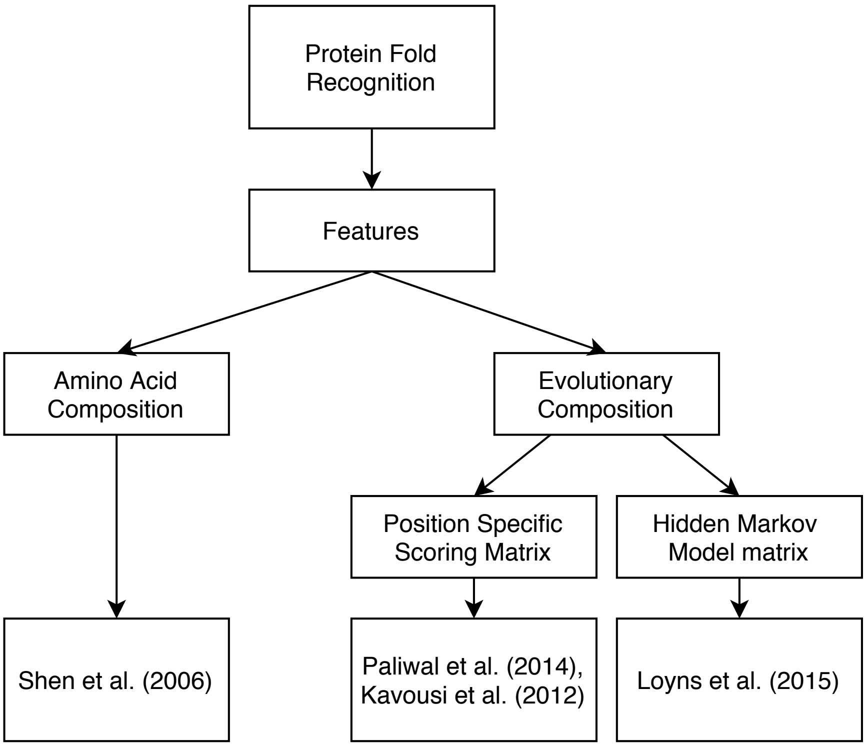


Figure 2.4. Categorization of state-of-the-art PFR models based on the feature source.

Table 2.3. Summary of State-of-the-art PFR Algorithms.

Methods	**Approach**	**Advantages**	**Disadvantages**
Shen and Chou Shen and Chou (2006)	PAAComposition, AAComposition based, Physicochemical-Based.	Fast and Less number of features. Accuracy increased to 70.5%.	Didn't consider important evolutionary information.
Paliwal et al. Paliwal *et al.* (2014*b*)	Tri-gram using PSSM	Considered all possible PSS based evolutionary features. Reached 71% accuracy for TG dataset.	HMM based evolutionary profiles were not considered during feature extraction.
Kavousi et al. Kavousi *et al.* (2012)	Predicted Secondary Structure, Physicochemical attributes and PSSM	Considered all possible evolutionary features. Reached 73.1% accuracy for DD dataset.	Limited physicochemical attributes (4) and PSSM (20) are considered. HMM based evolutionary profiles were not considered during feature extraction.
Loyns et al. Lyons *et al.* (2015)	mono, bi, tri-gram features from HMM Profile	HMM based evolutionary profiles were considered during feature extraction. Accuracy crossing 80% for benchmark datasets	Tri-gram features are higher in number (8000) and exhibits more overlapping information.

2.4 Protein Subcellular Localization

As per cellular anatomy, a cell (eukaryotic) is composed of different subcellular parts, or organelles Chou and Shen (2007). Protein performs its functions when it is located in appropriate compartments. Hence, it is very important to identify the subcellular location for the given protein. Prior information of the subcellular location of a given protein helps in i) understanding the protein function(s) ii) understanding about what other proteins are situated in the same compartment which helps in predicting interactions between proteins iii) finding biological process in which complex pathways are regulated at cellular level iv) developing appropriate drug to avoid side effects Chou and Shen (2008, 2010).

The chloroplast is one of the most classic organelles that are found in algae and plant cells. The primary function of chloroplast organelle is to conduct photosynthesis Melkikh *et al.* (2010). Besides, proteins in the chloroplast organelle play a vital role in various metabolic activities such as, amino acid synthesis Kirk and Leech (1972), immune response and lipid metabolism Wang and Benning (2012), pigment biosynthesis Moore and Shephard (1978), and fatty acid synthesis Post-Beittenmiller *et al.* (1992).

As per UniProt data bank[2], chloroplast organelle is mainly consisting of smaller structures or compartments such as envelope, plastoglobule, stroma, thylakoid lumen, thylakoid membrane. These compartments are at the sub-subcellular level and they exhibit a specific role in the overall chloroplast metabolism. The envelope compartment is made up of a double membrane. The plastoglobuli compartment acts as lipid reservoirs for thylakoid membranes. The stroma compartment is the inner chloroplast membrane excluding thylakoid space. The thylakoid lumen compartment is bounded by the thylakoid membrane and the thylakoid membrane compartment is responsible for the light reaction of photosynthesis.

Identification of proteins that are located in the sub-chloroplast compartments help in further understanding their roles in the various chloroplast biological activities. As per the Swiss-Prot database which is released in March 2007, only 20% of protein data from this database has been annotated with subcellular location information Chou and Shen (2007). Recent enhancements in large-scale sequencing technologies led to an increase in the deposition of a massive number of protein sequences in protein data banks. In 1986, Swiss-Prot Bairoch and Apweiler (2000) database had 3,939 protein sequences; whereas, now the number has grown to 560,537 according to the latest release by UniProtKB/Swiss-Prot UniProtKB/Swiss-Prot (2019). The number of protein entries is grown 142 times in the last 33 years. Biological-based experiments such

---

[2] `https://www.uniprot.org/locations/?query=chloroplast&sort=`.

as cell fractionation, fluorescence microscopy, and electron microscopy are utilized to predict PSCL. However, biological methods for a huge number of proteins to determine their localizations at sub-organelle (i.e., sub-subcellular) level is cost-intensive, time-consuming, and infeasible solution. Hence, there is a need to develop a fast, reliable, effective, and generalized computational model to identify PSCL.

Over the past decades, continuous efforts have been channelized and successfully developed various computational models to predict protein subcellular localization. These models can broadly categorize into three such as, Sequence-based, Evolutionary-based, and Knowledge-based approaches. Sequence-based approaches include: (i) amino acid composition-based models Chou (2001); Zhou and Doctor (2003); Fan and Li (2012); Dehzangi *et al.* (2015); Zhang and Duan (2018). The evolutionary-based approach makes use of profile information to predict Wan *et al.* (2016*b*). Knowledge-based approaches mainly rely on knowledge corpus such as PubMed data Brady and Shatkay (2008); Fyshe *et al.* (2008) and Gene Ontology (GO) Chou *et al.* (2011); Xiao *et al.* (2017); Cheng *et al.* (2018). The prediction of sub-chloroplast localization (i.e., sub-subcellular localization) is one-level more microscopic by nature compared to subcellular localization and hence it possesses a greater challenge in the prediction of sub-chloroplast localization compared to other subcellular localization problems.

Due to the higher difficulties in predicting PSCL, only a limited number of works have been proposed to solve the PSCL prediction problem. Earlier predictors such as, SubChlo Du *et al.* (2009), ChloroRF Tung *et al.* (2010), SubIdent Shi *et al.* (2011), and BS-KNN Hu and Yan (2012) are developed to solve single-label PSCL prediction problem. For a given query chloroplast protein, these single-label predictors assume that the query protein belongs to any one compartment of chloroplast organelle and hence they are able to identify only one sub-chloroplast location. However, a few chloroplast proteins are found to co-locate in multiple compartments of chloroplast organelle. For example, glyceraldehyde phosphate dehydrogenase Ferro *et al.* (2003) is found to be present in both stroma and envelope compartments of chloroplast organelle; Ferredoxin-NADP reductase Hanke *et al.* (2005) exists in both thylakoid membrane and stroma compartments of chloroplast organelle. Hence, the PSCL is considered to be a multi-label prediction problem. As these single-label PSCL predictors became ineffective in identifying multiple locations of chloroplast proteins, recent past multi-label predictors such as AL-KNN Lin *et al.* (2013), MultiP-SChlo Wang *et al.* (2015), are EnTrans-Chlo Wan *et al.* (2016*b*) are proposed.

Both AL-KNN Lin *et al.* (2013) and MultiP-SChlo Wang *et al.* (2015) predictors are sequence-based approaches and they adopt genetic algorithms to identify the relevant patterns from the pseudo amino acid composition (PseAAC). In AL-KNN Lin *et al.* (2013), the identified patterns are utilized by a multi-label K-nearest neighbor (KNN) classifier; whereas in MultiP-SChlo Wang *et al.* (2015), the identified patterns are utilized by multi-label support vector machine (SVM) to solve PSCL prediction. EnTrans-Chlo Wan *et al.* (2016*b*) predictor ensembles a relevant set of patterns from PseAAC and evolutionary profiles and these patterns are utilized by adopting the transductive learning approach which is based on least squares and K-nearest neighbor.

All the multi-label works of PSCL prediction are summarized in Table 2.4 and the categorization of these works based on the feature sources are shown in Figure 2.5.

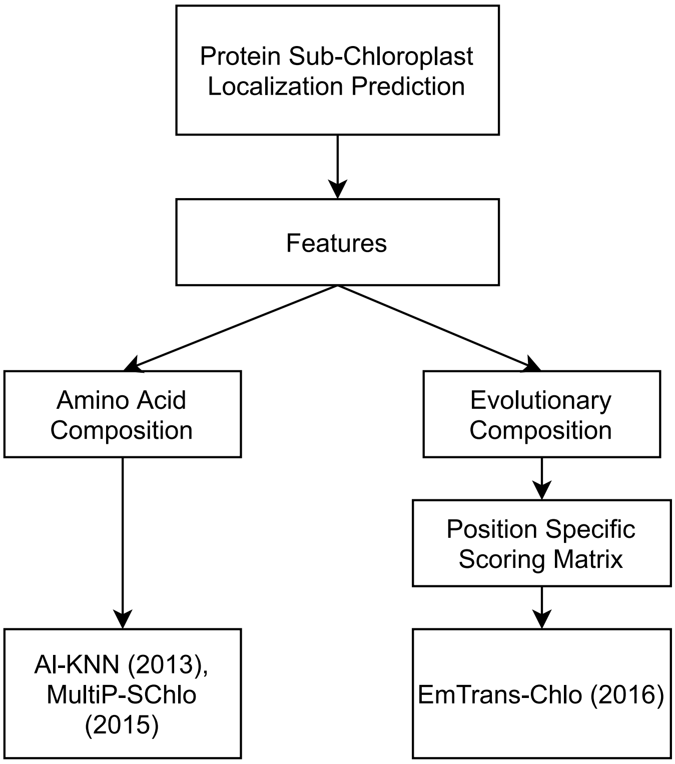

Protein Sub-Chloroplast Localization Prediction
Features
Amino Acid Composition
Evolutionary Composition
Position Specific Scoring Matrix
AI-KNN (2013), MultiP-SChlo (2015)
EmTrans-Chlo (2016)

Figure 2.5. Categorization of state-of-the-art PSCL models based on the feature source.

Table 2.4. Summary of State-of-the-art multi-label PSCL Algorithms.

Methods	**Approach**	**Advantages**	**Disadvantages**
Al-KNN Lin *et al.* (2013)	Pseudo amino acid composition features (PseAAC) with adaptive ML-kNN approach	Simple features are extracted. kNN is fast.	Unable to handle overlap features which exhibits different locations.
MultiP-SChlo Wang *et al.* (2015)	Pseudo amino acid composition features (PseAAC) with GA-based Binary Relevance (BR) framework for multi-label	Simple features are extracted. GA with BR is effective for multi-label.	Important PSSM features are missed in BR framework. More processing components are required.
EnTrans-Chlo Wan *et al.* (2016*b*)	Hybrid of PseAAC and PSSM features. Transductive approach with k-NN.	Quality set of features extracted. Reported 61% accuracy	Transductive approach doesnt work well for unseen data.

2.5 Research Gaps

Based on the extensive literature survey, the following key research issues and challenges have been identified for our proposed research work. The details are as follows:

- In the post-genome era, the rapid advancement of next-generation sequencing (NGS) techniques and high-end machines such as Illumina Oyola *et al.* (2012) generate a large number of nucleotide sequences in a given time Chowdhury and Garai (2017). These nucleotide sequences can be converted to protein codons in linear time with negligible error Lodish *et al.* (2000*b*), resulting in a large volume of protein data which are neither analyzed nor annotated Chowdhury and Garai (2017). In contrast to the sequencing techniques, experimental methods including sequence alignment to analyze these un-annotated sequences are time-consuming Chowdhury and Garai (2017). Thus, the difference between the rate at which raw sequences get stored in databases and the rate of their corresponding analysis for annotation is increasing rapidly Chowdhury and Garai (2017). Although, recent works on MSA concentrates on effective utilization of memory Khan *et al.* (2016), and exploring the parallelization aspects using external hardware components for pairwise alignment Fei *et al.* (2018) and using GPUs for MSA Blazewicz *et al.* (2013) to accelerate the MSA throughput. However, the time complexity of the MSA model plays a crucial role in speeding up the alignment, and these new developments in MSA can contribute significantly if the MSA time complexity is improved, thereby reducing the rate difference of storage and analysis of sequences. Hence, to address these challenges, we proposed a computationally efficient (polynomial time) MPSA model by adopting a progressive alignment approach, since both classic and iterative approaches exhibit higher-order time complexity. Further, to overcome the limitations of the progressive approach, we propose an SIO framework to improve the local optima trap.

  MSA is an important and primary step for many biological activities such as protein secondary structure class prediction, identification of protein folds, and protein subcellular location prediction, etc. The results from current methods for MSA are not satisfactory. Moreover, the existing model adopts a static gap penalty strategy in scoring alignment which leads to major shortcomings such as (i) MSA program demands expert input for providing pre-defined penalty values, (ii) penalty values are not consistent across various similarity datasets, and (iii) negative impact on MSA accuracy. Further, the empirical way of finding gap penalties for various sequence similarity datasets is not only tedious but also a time-consuming task. The authors of Kim and Kececioglu (2008) reported that the accuracy of an alignment can be improved by a factor of up to 25% using an effective scoring system. Also, it is evident from the reported works that there is further scope to explore the gap penalty strategy to perform a biologically more effective alignment.

- Secondary structure class prediction is one of the intermediate steps in the process of identifying the protein tertiary structure. The previous studies have revealed that the protein sequences, structural sequences, and evolutionary profile information provide promising ways to improve the effectiveness of the PSSC prediction. However, these studies lack in extracting generic features since they have mainly focused on either high or low similarity datasets. Moreover, a very limited or no study is been carried out to explore character embedding and skip-gram techniques to extract an effective set of features. Hence, there is ample scope to extract a generic set of features from both high and low sequence similarity datasets.

- Protein fold recognition is one step closer to predict the tertiary structure of a protein from its amino acid sequence. In literature, the best existing method to predict protein fold recognition has just crossed 80% accuracy. None of the existing methods explored convolutional with deep learning approaches to solve PFR. The amino acid sequence gets folded to form stable protein due to ionic interactions, hydrophobic bond(s), di-sulfide bond(s), and hydrogen bond(s). These amino acid interaction patterns can be captured effectively using a deep neural network.

- Evolutionary-based profile of a query protein exhibits rich evolutional information and various prediction problems of proteins are successfully solved using evolutionary-based approach Lyons *et al.* (2015); Kumar *et al.* (2020). A limited number of predictors explored the evolutionary-based approach to solve the PSCL problem and still there exists a scope of further exploration of evolutionary profiles in extracting discriminating patterns/features. To the best of our knowledge, none of the PSCL predictors explored deep neural networks to solve the PSCL prediction problem. Moreover, the prediction performances of all the state-of-the-art PSCL predictors (including single-label and multi-label) are very limited.

2.6 Problem Statement

“ To develop an effective framework for Computational Analysis of Protein Structure and its Subcellular Localization using Amino Acid Sequences ”

2.7 Research Objectives

1. To design and develop an effective multiple sequence alignment technique for annotating amino acid sequences.
2. To design and develop an effective technique for secondary structural class prediction for amino acid sequences.
3. To design and develop an effective technique for protein fold recognition.
4. To design and develop an effective technique for protein subcellular localization prediction for eukaryotic cells.

2.8 Summary

This chapter provided a review of existing state-of-the-art methods on MPSA, PSSC prediction, PFR prediction, and PSCL prediction. The problem statement and research objectives were framed based on the outcome of the literature review.

In the next chapter, the proposed progressive-based multiple protein sequence alignment will be discussed.

Chapter 3

Multiple Protein Sequence Alignment

This chapter [1] proposes an effective progressive-based multiple protein sequence alignment method incorporated with a novel scoring system and single iterative optimization framework.

3.1 General Problem Statement:

For a given set of n unaligned protein sequences S: $\{ps_1, ps_2, ..., ps_n\}$ of variable length L_1, L_2, ..., L_n respectively, unaligned protein sequences are defined over 20 amino acids' alphabet set Σ=$\{$*A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y*$\}$. MPSA for a given input sequence set S is defined as S': $\{ps'_1, ps'_2, ..., ps'_n\}$, where the length of all n aligned sequences are the same. S' is defined on the same alphabet set with an additional symbol '–' termed as a gap. An example of understanding MPSA is shown in Table 3.1. A gap (–) is introduced at required positions, not only to make all sequences with equal length but to ensure that the alignment is accurate. A gap is termed as indel (insertion-deletion) i.e. from the Table 3.1, a residue M is either inserted in ps'_2 or deleted in ps'_3. From ps'_1 and ps'_2 it can be observed that either a residue I is substituted with a residue M or vice versa over the period. These alterations of one or more residues such as indels or substitutions are considered as mutations.

3.2 Scoring System

The quality of MPSA is dependent on the scoring function, which aligns a given residue to its appropriate position. Three possible cases arise when the scoring function encounters a residue pair: (i) a residue with the same residue; (ii) a residue with a different residue; or (iii) a residue with a gap. For the first two cases, a heuristic-based scoring strategy named Look Back Ahead (LBA) scoring strategy is proposed. For the last case, Position-Residue Specific Dynamic Gap Penalty (PRSDGP) scoring strategy is proposed to find overall gap penalty, i.e., Affine Gap Penalty (AGP).

---

[1] The work described in this Chapter has been published in: **Sanjay Bankapur** and Nagamma Patil, "Efficient and Effective Multiple Protein Sequence Alignment Model Using Dynamic Progressive Approach with Novel Look Back Ahead Scoring System" in *Proceedings of the 7th International Conference on Pattern Recognition and Machine Intelligence 2017 (PReMI '17)*, ISI Kolkata, India, Dec-2017, Springer.
Sanjay Bankapur and Nagamma Patil, "Position-Residue Specific Dynamic Gap Penalty Scoring Strategy for Multiple Sequence Alignment" in *Proceedings of the 8th International Conference on Computational Systems-Biology and Bioinformatics (CSBio '17)*, Nha Trang, Vietnam, Dec-2017, ACM.
Sanjay Bankapur and Nagamma Patil, "ProgSIO-MSA: Progressive based Single Iterative Optimization Framework for Multiple Sequence Alignment using an Effective Scoring System" in *Journal of Bioinformatics and Computational Biology, World Scientific*. (May 2020).

Table 3.1. Example of MPSA

Input Unaligned Sequences	**Output Aligned Sequences**
ps_1: NIMGS (5)	ps'_1: NIMGS (5)
ps_2: NMMGS (5)	ps'_2: NMMGS (5)
ps_3: NFGS (4)	ps'_3: N– FGS (5)
ps_4: NLGS (4)	ps'_4: N– LGS (5)
ps_5: NHS (3)	ps'_5: N– – HS (5)

3.2.1 Look Back Ahead Scoring Strategy

Look Back Ahead (LBA) follows a heuristic approach, in which a given pair of residue is scored based on three things: (i) match or mismatch score of current residue pair; (ii) similarity score of all residue pair up to the previous residue; and (iii) status score of the previous residue.

We adopted Sum-of-pairs (SP) scoring function to calculate a similarity score for a given column from a sequence pair or from a profile pair, where a profile is a set of aligned sequences. SP scoring function for the aligned sequences is shown in equation 3.1.

$$SP(S') = \sum_{l=1}^{L} \sum_{i=1}^{n} \sum_{j=i+1}^{n} m(ps'_{i,l}, ps'_{j,l}) \tag{3.1}$$

where, S' is the aligned sequences, L is the length of the alignment, n is the total number of sequences, m is a mutation matrix, $ps'_{i,l}$ and $ps'_{j,l}$ are the residue of l^{th} column of i^{th} and j^{th} sequence respectively.

To calculate the similarity score for a given column from a sequence or a profile pair is as shown in equation 3.2.

$$ss_{i,j}(U, V) = \frac{\sum_{p=1}^{P} \sum_{q=1}^{Q} m(U_p, V_q)}{|U| * |V|} \tag{3.2}$$

where, $ss_{i,j}(U, V)$ is similarity score for i^{th} and j^{th} column of sequence or profile, U and V respectively. m is a mutation matrix, P, and Q are the number of sequences that exists in profile U and V respectively.

Given a pair of unaligned sequences, say ps_1 and ps_2, of length L_1 and L_2 respectively. Dynamic programming alignment approach generates a 2-dimensional dynamic matrix of size (L_1 + 1) * (L_2 + 1), where the extra row and column for initial default values and scores every cell in the matrix by exploring all possible cases to obtain an optimally aligned sequence pair. Each cell in the matrix is scored using LBA scoring strategy traversing left to right and top to bottom. The rightmost bottom cell provides a maximum similarity score for the given unaligned pair.

Let X be a matrix generated during dynamic programming. A score of i^{th} residue and j^{th} residue from sequence ps_1 and ps_2 respectively, is given in equation 3.3.

$$Score(X_{i,j}) = max \begin{cases} (Score(X_{i,j-1}) - gappenalty) - D \\ (Score(X_{i-1,j-1}) + ss_{i,j}(AAM_i, AAM_j)) - E \\ (Score(X_{i-1,j}) - gappenalty) - F \end{cases} \tag{3.3}$$

where, $Score(X_{i,j-1})$, $Score(X_{i-1,j})$ and $Score(X_{i-1,j-1})$ are the scores of previous cells i.e. left, top and top-left-diagonal respectively. *gappenalty* value is calculated using PRSDGP scoring strategy (which will be discussed in section 3.2.2), $ss_{i,j}(AAM_i, AAM_j)$ is the value obtained from equation 3.2, AAM is Amino Acid Molecule or residue, AAM_i is the i^{th} column residue from ps_1 and AAM_j is the j^{th} column residue from ps_2. Here *D, E* and *F* are scores which are calculated on previous residue status. Each previous cell can have any one status among *diagonal-non-gap (dng)*, *left-gap (lg)* and *top-gap (tg)*, where, *left-gap* indicates a gap in sequence ps_1, *top-gap* indicates a gap in sequence ps_2 and *diagonal-non-gap* indicates no gap in any of the sequences. *D, E* and *F* are defined as follows:

$$D = \begin{cases} 0 & \text{if } Status(X_{i,j-1}) = lg \\ gapopenpenalty & \text{otherwise} \end{cases} \tag{3.4}$$

$$E = \begin{cases} 0 & \text{if } Status(X_{i-1,j-1}) = dng \\ gapclosepenalty & \text{otherwise} \end{cases} \tag{3.5}$$

$$F = \begin{cases} 0 & \text{if } Status(X_{i-1,j}) = tg \\ gapopenpenalty & \text{otherwise} \end{cases} \tag{3.6}$$

where, *gapopenpenalty* are calculated using PRSDGP scoring strategy.

gappenalty: is a penalty when a gap occurs in either of the sequences at the current position.

gapopenpenalty: is a penalty when a gap occurs in either of the sequences at the current position and no gap at the previous position.

gapclosepenalty: is a penalty when there is no gap at the current position and there is a gap at the previous position in either of the sequences.

In all our experiments, *gappenalty*, *gapopenpenalty* & *gapclosepenalty* values are in the ratio of 1:4:1. Since *gapopenpenalty* inserts a gap in the alignment, it is penalized with higher penalty value compared to *gappenalty* and *gapclosepenalty*. *gappenalty* is penalized irrespective of its position and *gapclosepenalty* is also penalized with less penalty since there is strong match among the current residues. This heuristic ratio is determined empirically since it provides better results across datasets.

Once $Score(X_{i,j})$ is calculated (from equation 3.3), $Status(X_{i,j})$ will be set to *left-gap* or *diagonal-non-gap* or *top-gap* if $Score(X_{i,j})$ is equal to *A* or *B* or *C* respectively.

3.2.2 Position-Residue Specific Dynamic Gap Penalty Scoring Strategy

Numerous models adopt various heuristic scoring systems to score a match or a mismatch residue pair. However, as per our knowledge, none of the models explored a scoring strategy for penalizing gaps dynamically based on its position and residue information. Thus, this is the first attempt to explore the Position-Residue Specific Dynamic Gap Penalty (PRSDGP) scoring strategy in which it calculates gap penalty dynamically based on a residue and its position information using a biologically effective mutation matrix.

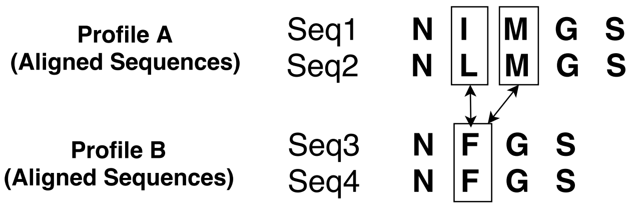
$$AGP(S') = \begin{cases} (gapopenpenalty \times \#opening\_gap) \\ + \\ (gapextendpenalty \times \#trailing\_gap) \end{cases} \tag{3.7}$$

Affine Gap Penalty (AGP) strategy to calculate the overall gap penalty of the alignment, and it is as shown in equation 3.7. Where, *#opening_gap* is the total number of opening gaps (a gap at the current position and non-gap at its previous position), *#trailing_gap* is the total number of trailing gaps (a gap at the current position as well as in its previous position), *gapopenpenalty* is the penalty value which is calculated using PRSDGP scoring strategy. In all our experiments, *gapopenpenalty* to *gapextendpenalty* ratio is considered to be 10:1.

PRSDGP score for a given residue pair is defined as the average of mutation scores of possible occurrences of both the residues and it is described as in equation 3.8, where, m is a mutation matrix and AAM is a amino acid molecule.

$$PRSDGP(AAM_i, AAM_j) = \begin{cases} (\frac{m(AAM_i, AAM_i)}{2}) \\ + \\ (\frac{m(AAM_j, AAM_j)}{2}) \end{cases} \tag{3.8}$$

The PRSDGP score between two profiles (aligned sequences) for a given position is calculated as an average of the sum of all pair-wise PRSDGP scores. As shown in Figure 3.1, four sequences are aligned in two profiles, A and B, in which PRSDGP score for the second column of both the profiles, PRSDGP(A2, B2), is calculated using equations shown in Figure 3.1 and 3.8 i.e. ((5.0) + (5.0) + (5.0) + (5.0))/4 = 5.0. Similarly, for the third column of profile A and the second column of the profile B, PRSDGP(A3, B2), is calculated i.e. ((5.5) + (5.5) + (5.5) + (5.5))/4 = 5.5. To calculate the PRSDGP scores, the BLOSUM62 mutation matrix is used. The objective is to minimize the gap penalty; therefore, with these PRSDGP scores, we can say that aligning the second column of profile B to the third column of profile A is more appropriate instead of the second column of profile A. Moreover, the SP score for aligning the second column of profile B to the second and third column of profile A respectively, are zeroes. Hence, the PRSDGP score is a major deciding factor for alignment.



PRSDGP(A2, B2) = [PRSDGP(I, F) + PRSDGP(L, F) + PRSDGP(I, F) + PRSDGP(L, F)] / 4

PRSDGP(A3, B2) = [PRSDGP(M, F) + PRSDGP(M, F) + PRSDGP(M, F) + PRSDGP(M, F)] / 4

Figure 3.1. Calculation of PRSDGP scores for two aligned profiles.

3.3 Progressive Alignment Method using LBA and PRSDGP Scoring Strategies

A basic progressive alignment approach consists of three main steps: (i) calculation of pair-wise similarity score matrix for all possible sequence pairs; (ii) generation of guide tree from similarity score matrix; and (iii) alignment of sequences based on branching order of the generated guide tree.

During progressively aligning multiple sequences, three possible cases arise: (i) alignment of a sequence with another sequence (S-S); (ii) alignment of a sequence with a profile (S-P) or a profile with a sequence (P-S); and (iii) alignment of a profile with a profile (P-P), where, profile is a set of sequences which are already aligned. The advantage of adopting local alignment for pairwise progressive alignment is that the consistency in identifying the homologous core blocks and aligning is considerably high when compared to global alignment, especially for low sequence similarity datasets. The steps to generate intermediate aligned result from n unaligned amino acid sequences using progressive alignment approach are shown in Algorithm 3.1 and the workflow for the algorithm is depicted in Figure 3.2.

Time complexity Analysis: Given n unaligned protein sequences of variable length. Let L be the maximum length from the given sequences. Time complexity to compute pair-wise alignment for nC_2 combinations is O($n^2.L^2$) and to build guide tree using UPGMA strategy Sneath and Sokal (1973) for n protein sequences is O(n^2). Time

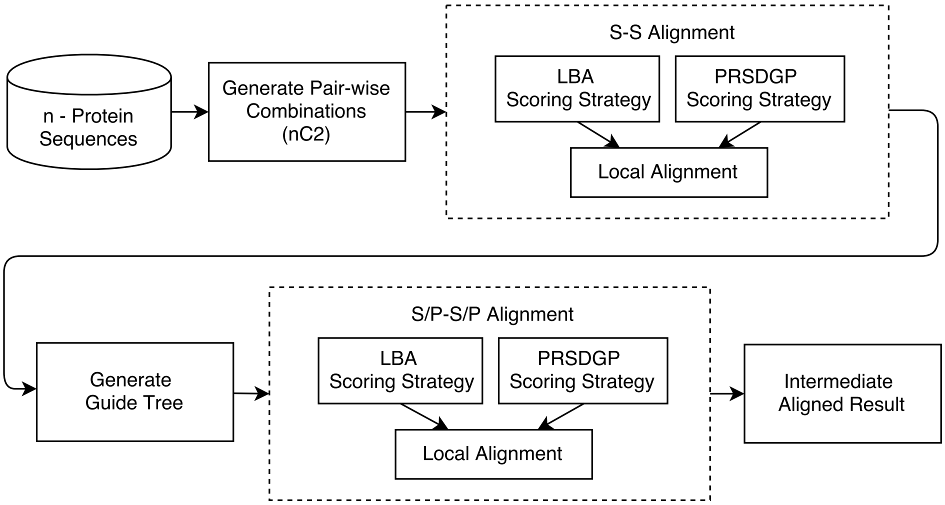


Figure 3.2. Progressive alignment using LBA and PRSDGP scoring strategies.

Algorithm 3.1. : Progressive Alignment using Proposed LBA and PRSDGP Scoring Strategies

```
Input: $n$ unaligned protein sequences of variable length, where $n \geq 2$.
Output: Aligned $n$ protein sequences of fixed length, say $L$.
1: For all possible $nC_2$ combinations, find a similarity score matrix by performing
   pair-wise Local Alignment using the proposed scoring system.
2: Generate guide tree using UPGMA strategy from the similarity score matrix.
3: Local alignment of sequences with LBA and PRSDGP scoring strategies based on
   branching order of generated guide tree.
```

complexity to align n sequences progressively using guide tree is O(n^3 + $n.L^2$). Hence, the time complexity to obtain intermediate alignment result using progressive approach is O($n^2.L^2$) + O(n^2) + O(n^3 + $n.L^2$) which is equal to O(n^3 + $n.L^2$).

3.4 Single Iterative Optimization Framework to improve Local Optima

Due to the greediness of the progressive approach, the intermediate result may tend to suffer from local optima. To address this limitation, we proposed a Single Iterative Optimization (SIO) framework to identify and optimize the local optima regions using the proposed scoring system.

A. Objective Functions

It has been observed from recent works on optimization of MSA that maximizing the Sum-of-Pairs (SP) score and minimizing the Affine Gap Penalty (AGP) score, tend to improve the quality of the alignment result Naznin *et al.* (2012), Zhu *et al.* (2016), Rubio-Largo *et al.* (2016). Therefore, in this study, to identify and optimize the local optima ranges, we considered SP and AGP as the objective functions. SP and AGP are defined as shown in equations 3.1 and 3.7 respectively.

B. To Identify Local Optima Ranges

Local Optima Range (LOR) is a column or set of consecutive columns of the alignment for which the quality is poor, and it is identified using defined objective functions, i.e., SP and AGP. The proposed steps to identify a LOR are as follows:

1. Calculation of GSP and GAGP Scores: From the intermediate aligned result, SP and AGP scores which are calculated using equations 3.1 and 3.7 respectively are termed as Global SP (GSP) and Global AGP (GAGP) scores.

2. Normalization of GAGP: Normalize GAGP score with respect to GSP score by a factor of α, i.e., α units of GAGP for one unit of GSP, as shown in equation 3.9.

$$GSP = \alpha.GAGP \tag{3.9}$$

3. Calculation of mean and standard deviation of GSP and GAGP scores: mean and standard deviation of GSP score with reference to all columns are calculated from the aligned result, named as mGSP and sdGSP respectively, and are defined in equations 3.10 and 3.11 respectively.

$$mGSP = \frac{GSP}{L} \tag{3.10}$$

$$sdGSP = \sqrt{\frac{1}{L}\sum_{i=1}^{L}(SP_i - mGSP)^2} \tag{3.11}$$

As LOR is exhibited by either a column or set of columns and AGP score involves row-wise calculation. Therefore, to calculate the mean and standard deviation of all row-wise AGP scores (i.e., mGAGP and sdGAGP respectively), we make use of the normalized value of GAGP regarding GSP, and it is as shown in equations 3.12 and 3.13.

$$mGAGP = \frac{mGSP}{\alpha} \tag{3.12}$$

$$sdGAGP = \frac{sdGSP}{\alpha} \tag{3.13}$$

4. Identification of Local Optima Range: Each column from the intermediate aligned result possesses SP score and a column SP score is termed as Local SP (LSP). Similarly, for each column, the AGP score is calculated and it is termed as Local AGP (LAGP). On the obtained intermediate aligned result, a single iteration is performed to find local optima columns by scanning column-wise from left to right. LOR is identified by the two steps: (i) if LSP score is lower than mGSP and LAGP score is higher (since penalty scores should be minimized, a higher score indicates poor alignment quality and lower scores indicates better alignment quality) than mGAGP, then the column is marked as a candidate for LOR; and (ii) consecutive three or more columns are considered as candidates for LOR then they are added to the LOR list if the sdLSP is less than the sdGSP and sdLAGP is greater than sdGAGP.

Let L be the length of intermediate alignment and length of an LOR be denoted as r_L, which ranges from ϕ to ψ, i.e. $\phi \leq r_L \leq \psi$. In this study, we considered $\phi \geq 3$ and $\psi \leq$ L. There can be K number of LORs in the LOR list, where, K varies from 0 to $\lfloor(\frac{L}{\phi} - 1)\rfloor$. By this, we can deduce that both K and r_L are inversely proportional, i.e., as the length of a LOR increases, the number of LORs in the LOR list decreases. The proposed algorithm to identify the LOR is shown in Algorithm 3.2 and the summarized workflow is as shown in Figure 3.3.

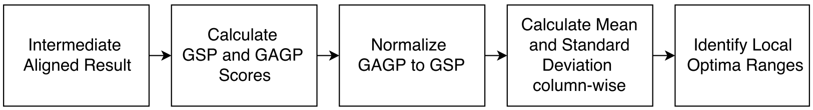

Intermediate Aligned Result
Calculate GSP and GAGP Scores
Normalize GAGP to GSP
Calculate Mean and Standard Deviation column-wise
Identify Local Optima Ranges

Figure 3.3. Steps to identify local optima ranges.

Algorithm 3.2. : Proposed Algorithm to Identify Local Optima Ranges

Input: n aligned amino acid sequences.

Output: List of local optima ranges.

```
Calculate GSP and GAGP scores from the given aligned sequences.
Normalize GAGP score with respect to GSP score by a factor of $\alpha$.
Calculate mean and standard deviation of GSP score with reference to all columns
  – mGSP and sdGSP.
Calculate mean and standard deviation of GAGP score with reference to all columns
  – mGAGP and sdGAGP (in terms of GSP and $\alpha$).
 for all columns of aligned sequences $\{c_1, c_2, c_3, ..., c_L\}$
    each column $c_i$, Calculate LSP and LAGP scores.
    if (LSP $\geq$ mGSP && LAGP $\leq$ mGAGP) then
      skip the column $c_i$.
    else
      column $c_i$ is considered as a candidate for LOR.
 end for
Standard deviation of LSP and LAGP for three consecutive or more candidate
  columns of LOR be sdLSP and sdLAGP respectively. Selected LOR is added to
  LOR list only if $sdLSP < sdGSP$ and $sdLAGP > sdGAGP$.
```

Time complexity analysis: Time complexity to identify a list of LORs is approximately O(n^3), as complexity to calculate GSP score is O(n^3) and the rest of the calculations are relatively minimal.

C. Applying Progressive Alignment Method using LBA and PRSDGP Scoring Strategies for LORs

The identified list of LORs is optimized using the progressive approach with LBA and PRSDGP scoring strategies and the performed steps are shown in the Algorithm 3.3. The workflow of the algorithm is as shown in Figure 3.4.

Algorithm 3.3. : Optimization of LORs using Progressive Approach with LBA and PRSDGP Scoring Strategies

```
Input: $K$ number of LORs and intermediate aligned result.
Output: Final optimal aligned sequences.
1: From the $K$ LORs, select a LOR.
2: All the columns from a LOR, calculate LSP and LAGP scores are calculated and stored as oldLSP and oldLAGP respectively.
3: The LOR aligned data is pre-processed by removing gap symbol (–) to make unaligned sub-sequences.
4: Generate pair-wise combinations ($nC_2$) for unaligned sub-sequences.
5: For each pair-wise combination, align using both Global and Local Alignment techniques with LBA and PRSDGP scoring strategies and the higher similarity score among Global and Local Alignment is considered.
6: Out of the better $nC_2$ similarity scores, select the pair with the highest similarity score as the best pair and merge them to make a single new profile.
7: Replace the merged pair with the new profile in the subsequence list, reducing the total sub-sequences count by one.
8: Until the subsequence count is one, go back to step 4.
9: Calculate new LSP and new LAGP for the new alignment – newLSP and newLAGP respectively.
10: If newLSP and newLGAP scores are better than oldLSP and oldLGAP scores respectively, the final new profile is merged by replacing the LOR data, else, the final new profile is ignored.
11: Repeat from step 1 until all LORs are optimized.
```

It can be observed from steps 5, 6, and 7 that the proposed SIO model is a dynamic variant of progressive alignment approach in which the best-selected sequence pair is replaced by its merged alignment to reduce the total number of sequences count by one and further the next best sequence pair is selected and processed. To overcome the local optima problem effectively, the best SP and AGP score alignment is chosen from the output of local and global alignment. In the proposed model, PAM250 and BLOSUM62 mutation matrices are used for global and local alignment respectively.

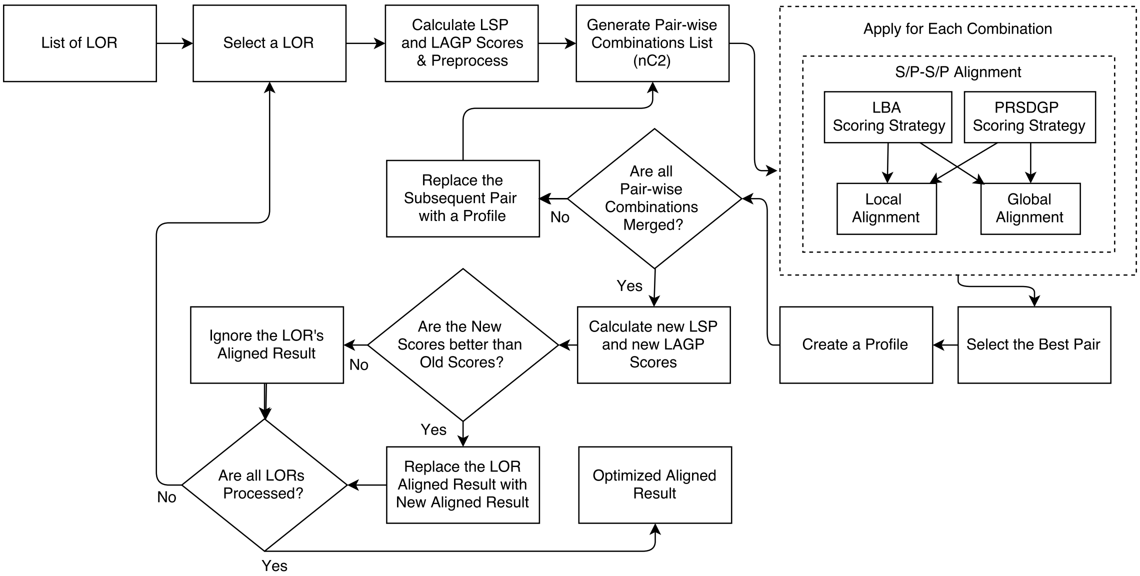

List of LOR
Select a LOR
Calculate LSP and LAGP Scores & Preprocess
Generate Pair-wise Combinations List (nC2)
Apply for Each Combination
S/P-S/P Alignment
LBA Scoring Strategy
PRSDGP Scoring Strategy
Local Alignment
Global Alignment
Replace the Subsequent Pair with a Profile
No
Are all Pair-wise Combinations Merged?
Yes
Create a Profile
Select the Best Pair
Ignore the LOR's Aligned Result
No
Are the New Scores better than Old Scores?
Calculate new LSP and new LAGP Scores
Yes
No
Are all LORs Processed?
Replace the LOR Aligned Result with New Aligned Result
Optimized Aligned Result
Yes

Figure 3.4. Workflow of single iterative optimization for local optima ranges.

Time complexity Analysis: Consider K number of LORs, each having length, r_L. The time complexity for execution of alignment using the dynamic variant of progressive approach, i.e., steps 3 to 7, are O(n) times and each time progressive alignment (both Local and Global) takes O(nC_2 . r_L^2). Hence, for all LORs, it takes O(K . $n - 1$. nC_2 . r_L^2). Considering the fact that K and r_L is inversely proportional, if r_L is maximum, then K will be minimum (equal to constant) value. Therefore, the overall time complexity of the SIO framework is approximately O(n . nC_2 . r_L^2) i.e., O(n^3 . r_L^2) which is of polynomial time.

3.4.1 Total Time Complexity Analysis of the Proposed Model

The total time complexity of the proposed model involves three major activities: (i) Obtaining intermediate aligned results using progressive alignment approach with LBA and PRSDGP Scoring Strategies [O(n^3 + n . L^2)], (ii) Identifying the list of LORs [O(n^3)] and (iii) Optimizing all LORs by dynamic variant of progressive approach [O(n^3 . r_L^2)]. Since all the three activities are performed sequentially in which, the value of r_L is one-third of L in the worst case and for the best case r_L value will be one, hence, the total time complexity of the proposed ProgSIO-MSA (Progressive-based Single Iteration Optimization - Multiple Sequence Alignment) model is O(n^3 + n . L^2), which is of polynomial time. Therefore, we can say that the proposed ProgSIO-MSA model is computationally efficient.

3.5 Results and Discussion

In this section, initially, we defined the evaluation metrics to assess the proposed ProgSIO-MSA model, followed by the overall experimental setup, and finally, we showcase the results with the analysis.

3.5.1 Evaluation Metrics

We considered three most commonly used metrics to access the aligned results and those are: (i) *Sum-of-Pair (SP)* Ortuno *et al.* (2012), Cutello *et al.* (2011), Rubio-Largo *et al.* (2016), Zhu *et al.* (2016): defined in equation 3.1; (ii) *Total Gap Penalty (TGP)* Kaya *et al.* (2014), Rubio-Largo *et al.* (2016), Zhu *et al.* (2016): defined in equation 3.7; and (iii) *Conserved Column Score (CCS)* Edgar (2004*b*), Rubio-Largo *et al.* (2016): defined by the number of columns in which each column residues are identical.

Even though TGP depicts the biological evolution process, we infer that it is not sufficient to conclude the measured alignment. This is mainly due to the two major factors: (i) Individual value for *gapopen* and *gapextend* is not universally the same Naznin *et al.* (2012), Kaya *et al.* (2014), Rubio-Largo *et al.* (2016). (ii) The ratio between these two values is still debatable Naznin *et al.* (2012), Kaya *et al.* (2014), Rubio-Largo *et al.* (2016). Hence, we propose one more metric called Total Mutation Count Pair-wise (TMCP) to evaluate the proposed model result and its respective reference alignment. TMCP is defined as:

$$TMCP(S') = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} MC(ps'_i, ps'_j) \tag{3.14}$$

$MC(ps'_i, ps'_j)$ is total number of mutations for a given aligned protein sequence pair i.e., ps'_i and ps'_j. A column or set of consecutive columns are considered to be one mutation, only if respective column residues are not identical.

Table 3.2. Example of mutation count for a sample sequence pair

ps'_1	R	A	T	K	N	F	A	G	V	K	N
ps'_1	-	A	T	E	-	C	A	G	V	-	-
MC	1	0		1			0			1	

For example: From Table 3.2, ps'_1 and ps'_2 are two aligned protein sequences, MC is mutation count for the given pair; hence, the total mutation count for the given pair is 3. Lower the value of TMCP, the better is the alignment. There can be more than one alignment with the same set of SP, AGP, CCS, and TMCP scores for a given set of unaligned sequences. Therefore, *Q-Score (Q)* Thompson *et al.* (1999) and *Total Column Score (TC)* Thompson *et al.* (1999) are widely used evaluation metrics to assess the quality of aligned results of proposed ProgSIO-MSA model with respect to aligned reference set.

Q-Score: A score describes the similarity of the aligned result with respect to its reference set. All the reference sets are manually aligned by the experts based on various factors like amino acid composition, physicochemical properties, and many more. *Q-Score* values range from 0 to 1, when *Q-Score* = 1 indicates the aligned result set is exactly same as in the aligned reference set. Higher the *Q-Score* value, better is the biological accuracy (quality) of the alignment model.

Consider an aligned result of n protein sequences of length L. All the aligned residues from the i^{th} column is denoted as $M_{i,1}$, $M_{i,2}$, ..., $M_{i,n}$. For every residue pair, we define p_{ijk} such that p_{ijk} = 1 if the residues $M_{i,j}$ and $M_{i,k}$ are aligned to each other with respect to the aligned reference set; else, p_{ijk} = 0. The Q-Score for i^{th} column is q_i is defined as follows:

$$q_i = \sum_{j=1}^{n} \sum_{k=j+1}^{n} p_{ijk} \tag{3.15}$$

Q-Score (Q) for the aligned result is defined as:

$$Q = \frac{\sum_{i=1}^{n} q_i}{\sum_{i=1}^{n_r} qr_i} \tag{3.16}$$

Where, n_r is the number of columns from the aligned reference set and qr_i is the score q_i for the i^{th} column in the aligned reference set.

Total Column Score (TC): Let C_i be the column score for the i^{th} column, C_i = 1 if all the residues in the i^{th} column are aligned in the reference set, else, C_i = 0 and La be the length of the alignment. The *TC* for the aligned result is defined as:

$$TC = \frac{\sum_{i=1}^{n} C_i}{L_a} \tag{3.17}$$

From a given n sequences, even if one sequence from the aligned result is different with respect to the reference set, then the TC score for the aligned result will be zero, even though the rest $(n-1)$ sequences are aligned the same with respect to the reference set. From this scenario and the definitions of Q-Score and TC Score, we can infer that *TC* = 0 if *Q-Score* = 0, but the converse need not be true. By this, we can say that even misplacement of single AAM by the alignment method with respect to reference alignment, the *TC Score* might drastically reduce (even to zero). TC Score does not provide the actual quality information and hence, it is not a good metric to assess MSA quality. In this study, we choose only the Q-Score metric as a quality metric to assess the proposed ProgSIO-MSA model.

3.5.2 Experiment Setup

A. Runtime Environment

The experiments are conducted in a workstation with Intel(R) Core(TM) i7 3.60GHz octa-core 64-bit CPU running on Ubuntu 16.10 with 16GB RAM. The proposed ProgSIO-MSA model is implemented in Java, Eclipse Platform 3.8.1.

B. Datasets

In this comparative study, the experimental analysis is conducted on two benchmark datasets to assess the performance of our proposed ProgSIO-MSA model: BAliBASE Bahr *et al.* (2001) and SABmark Van Walle *et al.* (2004*b*). The main characteristics of both datasets are summarized below:

1) *BAliBASE*: This is one of the widely used reference datasets. BAliBASE 2.0 Bahr *et al.* (2001) in which the reference alignments are categorized into multiple subgroups based on the sequence characteristics. We considered the first five subgroups of version 2.0 those are: (i) Ref.1: exhibits similar length for a small number of equidistant sequences with no large insertions. (ii) Ref.2: both closely related sequences, i.e., (sequences identity $> 25\%$) and orphan sequences which are less than 20% identity are grouped in this category. (iii) Ref.3: equidistant divergent families (up to four families) are aligned in each set and the sequence identity is lesser than 25% among the different family sequences. (iv) Ref.4: sequences exhibit a large N/C terminal extension and (v) Ref.5: exhibits large internal insertions and deletions.

2) *SABmark*: aligned reference set are categorized into two subgroups: (i) Superfamily: consisting of 315 sequence data files in which the sequence identity ranges up to 50%. (ii) Twilight: consisting of 108 sequence data files in which it shares low sequence identity, i.e., 0-20%.

3.5.3 Results and Analysis

The results of the proposed ProgSIO-MSA model are initially analyzed using SP, TGP, CCS, TMCP scores. The alignment quality of the proposed ProgSIO-MSA model is analyzed using the Q-Score metric on the progressive-based and stochastic-based state-of-the-art models. Further, the scalability analysis of the proposed ProgSIO-MSA model against CLUSTAL Omega is performed. Finally, the statistical significance analysis is carried out for Q-Score results of BAliBASE.

A. Analysis of SP, TGP, CCS and TMCP scores:

To analyze the relationship between respective scores for test and reference sets, we performed the Pearson Correlation Coefficient Test Sedgwick (2012) for both BAliBASE and SABmark datasets. The value of the correlation ranges from -1 to +1 indicating the strength of the relationship among the samples, where, +1 indicates a positive correlation, -1 indicates a negative correlation and a zero value indicates no relationship among the samples.

Table 3.3. Pearson Correlation Coefficient Test on Aligned and Reference Scores

Dataset	**SP***	**TGP***	**CCS***	**TMCP***
BAliBASE	0.99940	0.95780	0.99800	0.99790
SABmark	0.99920	0.90742	0.93205	0.99242

* Scores are calculated up to five decimal point.

From Table 3.3, it is evident that, for all the four metrics, the respective scores for the test set and reference set, have correlated positively with the value above 0.9, inferring that the proposed scoring system is closely inclined with the reference sets.

B. Q-Score Analysis:

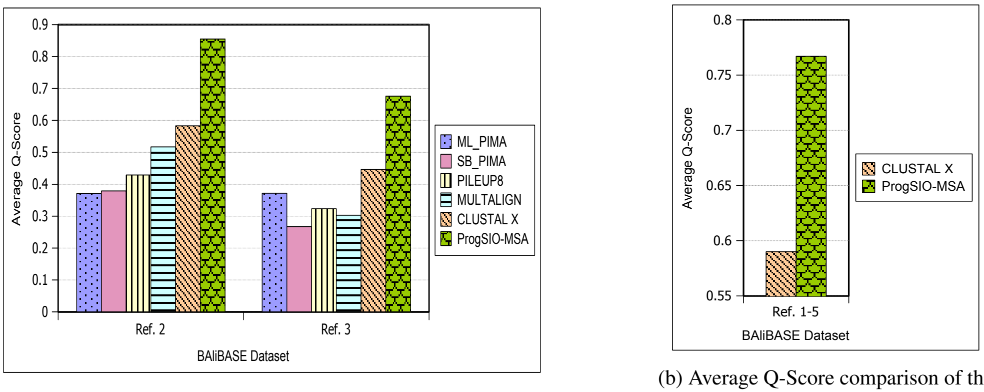
To assess the biological accuracy (quality) of the proposed ProgSIO-MSA model, we calculated Q-Score on both BAliBASE and SABmark datasets using the q-score program by Edgar. The proposed ProgSIO-MSA model being the progressive approach, it has been evaluated against by comparing progressive models. Further, the proposed ProgSIO-MSA model is also evaluated against progressive-based iterative models which are the most popular and effective models till date.

i) Analysis of Progressive-based models:

The performance of the proposed ProgSIO-MSA model is assessed on the BAliBASE dataset and compared with all the progressive models. The results are as shown in Table 3.4. The data files, progressive models, and their respective Q-Score results are referred from Naznin *et al.* (2012). The blank cell (-) indicates the unavailability of the results. Q-Score values marked bold signifies the best score among other models for the given subset. The Q-Scores from Table 3.4 are average scores. Both Ref.2 and Ref.3 subsets include all range of sequence similarities, i.e., from below 20% to above 25% and it can be observed from Table 3.4 that the performance of the proposed ProgSIO-MSA model on both subsets outperforms the other progressive based models by a factor of at least 25.9% collectively. For the overall Q-Score, the proposed ProgSIO-MSA model outperforms CLUSTAL X by a factor of 17.7% on the BAliBASE dataset.

Table 3.4. Average Q-Score comparison of the Proposed ProgSIO-MSA Model against the State-of-the-art Progressive Models on BAliBASE Dataset.

BAliBASE							
Models	**Ref.1**	**Ref.2**	**Ref.3**	**Ref.4**	**Ref.5**	**Ref.2&3**	**Overall**
ML_PIMA	-	0.371	0.372	-	-	0.371	-
SB_PIMA	-	0.379	0.267	-	-	0.342	-
PILEUP8	-	0.429	0.323	-	-	0.394	-
MULTALIGN	-	0.517	0.303	-	-	0.447	-
CLUSTAL X	0.708	0.583	0.446	0.315	0.689	0.538	0.590
ProgSIO-MSA	**0.752**	**0.855**	**0.676**	**0.433**	**0.735**	**0.797**	**0.767**



(a) Average Q-Score comparison of the proposed ProgSIO-MSA model against other progressive models on Ref.2 and Ref.3 subsets of BAliBASE

(b) Average Q-Score comparison of the proposed ProgSIO-MSA model against CLUSTAL X on BAliBASE dataset

Figure 3.5. Average Q-Score comparison on BAliBASE dataset

It is evident from Figure 3.5a and 3.5b that the proposed ProgSIO-MSA model outperformed the progressive models for BAliBASE dataset. To evaluate the performance consistency of the proposed ProgSIO-MSA model, we have assessed the proposed model on another benchmark dataset named SABmark dataset.

Table 3.5. Average Q-Score Comparison of the Proposed ProgSIO-MSA Model against Progressive based Models on SABmark dataset.

Dataset →	**SABmark**		
Models ↓	**SuperFamily**	**Twilight**	**Overall**
Align-m	0.445	0.172	0.375
CLUSTAL X	0.472	0.248	0.414
CLUSTAL O	0.478	**0.257**	0.421
ProgSIO-MSA	**0.508**	0.229	**0.436**

The performance of the proposed ProgSIO-MSA model is compared against the popular and widely used state-of-the-art models of CLUSTAL such as CLUSTAL X M.A. Larkin (2007) and CLUSTAL Omega Sievers *et al.* (2011). The latest versions of CLUSTALs are considered (CUSTAL Omega[2] version 1.2.4, CLUSTAL X[3] version 2.1). For the Align-M, we have referred to the Q-Score results from Edgar (2004*b*). The average Q-Scores of SABmark datasets for all these state-of-the-art models is shown in Table 3.5. Bold-faced values signify the best among other models for the given subset. From Table 3.5, it can be observed that the proposed ProgSIO-MSA model outperformed all the other models by a factor of at least 1.5%.

ii) Analysis of Stochastic-based models:

Over the last decade, stochastic-based iterative approaches are effectively explored to optimize and improve the alignment quality with the trade-off of higher space and time complexity. The proposed model being a progressive-based approach is further evaluated against selected popular genetic algorithm based MSA. A genetic algorithm is one of the effective stochastic-based optimization techniques which simulates the natural evolution process to optimize MSA. Genetic algorithm based MSA models usually follows three major steps: 1) *Initial population:* generates a set of random populations (generally 40 and above) and each population represents an alignment for a given set of sequences; 2) *Selection & New Generation:* the best pair of the population is selected to breed a new generation. In MSA, to improve the objective function will be considered as a fitness function for selection and genetic operators like Crossover and Mutation utilizes the breed to obtain the next generation. The new generation will replace the weakest population mimicking Charles Darwin's principle "survival of the fittest"; 3) *Termination:* a stopping criterion to terminate the generational process, which usually will be either based on solution convergence or for a fixed number of generation. From this, it is evident that the genetic algorithm based alignment models consume considerable space (for population, say P) and time (fitness evaluation for each generation, i.e., $O(n^3)$ for sum-of-pair calculations and over several generations, say G) to optimize the MSA quality. The most popular genetic algorithm based MSA models are Notredame and Higgins (1996), Gondro and Kinghorn (2007), Taheri and Zomaya (2009) and Naznin *et al.* (2012).

The average Q-Score of the proposed ProgSIO-MSA model and other state-of-the-art genetic algorithm based MSA models for the BAliBASE dataset is as shown in Table 3.6. The data files, a genetic algorithm based MSA models, and their respective Q-Score results are referred from Naznin *et al.* (2012). The blank cell (-) indicates the unavailability of the results. Q-Score values marked bold signifies the best score among other models for the given subset. From Table 3.6, it can be observed that the proposed

---

[2] `http://www.clustal.org/omega/`

[3] `http://www.clustal.org/clustal2/`

Table 3.6. The Q-Score Performance Comparison of the Proposed ProgSIO-MSA Model against Selected Genetic Algorithm based MSA Models on BAliBASE dataset.

BaliBASE						
Models	**Ref.1**	**Ref.2**	**Ref.3**	**Ref.4**	**Ref.5**	**Overall**
MSA-GA	0.655	-	-	0.374	0.475	-
MSA-GA w/prealign	0.730	-	-	0.334	0.675	-
SAGA	-	0.586	0.506	-	-	-
RBT-GA	-	0.777	0.472	-	-	-
GAPAM	**0.779**	0.851	0.662	0.208	**0.843**	**0.767**
ProgSIO-MSA	0.752	**0.855**	**0.676**	**0.433**	0.735	**0.767**

model outperforms genetic algorithm based MSA models except GAPAM Naznin *et al.* (2012). The proposed ProgSIO-MSA model outperformed the GAPAM model on three reference datasets (i.e., Ref.2, Ref.3, and Ref.4) of the BAliBASE dataset. The average Q-Score of the proposed ProgSIO-MSA model performs equally good when compared to the GAPAM model. In spite of running for multiple generations with higher alignment populations, GAPAM performs on par with the proposed model which runs for only one iteration.

iii) Computational Analysis:

The computational efficiency of the proposed ProgSIO-MSA model is analyzed against both progressive and stochastic-based iterative models. Initially, the computational efficiency analysis is carried out in terms of running time against the progressive-based iterative model, and later, it is carried out in terms of time complexity for the stochastic-based iterative model.

Table 3.7. Run Time Comparison of the Proposed ProgSIO-MSA Model against CLUSTAL Omega

Dataset		**CLUSTAL Omega in milli secs**	**ProgSIO-MSA in milli secs**
SABmark	Superfamily	104043	99000
	Twilight	40020	18000
	Overall	144063	**117090**

The proposed ProgSIO-MSA model being a progressive-based approach was compared against the running times of progressive-based iterative model, i.e., CLUSTAL Omega Sievers *et al.* (2011), since CLUSTAL Omega is one of the fast, scalable models which can align any number of sequences in a few hours Sievers *et al.* (2011). We recorded both the models' alignment running time on SABmark datasets and it is as shown in Table 3.7. From Table 3.7, we can observe that the proposed ProgSIO-MSA model took considerably less time to align all 423 files when compared to CUSTAL

Omega.

We compared the proposed ProgSIO-MSA model's time complexity against the best stochastic-based iterative model, i.e., GAPAM Naznin *et al.* (2012). GAPAM Naznin *et al.* (2012) being a genetic algorithm based iterative approach, constructs initial alignment population of size say 'P' using a progressive approach. To construct initial alignment population takes O($P.n^3$+$n.L^2$). These 'P' populations are improved using various operations over the 'G' number of generations to obtain optimum alignment. In each generation, based on fitness measure (i.e., the weighted sum of the pair which takes O(n^3)) of each alignment, a non-fit alignment is replaced by a fit alignment in the population pool. Hence, the time complexity of GAPAM Naznin *et al.* (2012) would be of O($G.P.n^3$+$n.L^2$).

Table 3.8. Time Complexity Comparison of the Proposed ProgSIO-MSA Model against Iterative Model

Models	**Approach**	**Time Complexity**
ProgSIO-MSA	Progressive	O($n^3 + n \;.\; L^2$)
GAPAM Model	Iterative	O($G \;.\; P \;.\; n^3 + n \;.\; L^2$)

The GAPAM time complexity and the proposed ProgSIO-MSA model time complexity (from the section 3.4.1) is as shown in Table 3.8. From the Table 3.8, it is evident that the proposed ProgSIO-MSA model is computationally efficient and outperforms by a factor of [$G.P$] when compared to the best genetic algorithm based stochastic model, i.e., GAPAM Naznin *et al.* (2012).

C. Scalability Analysis:

From both the benchmark datasets, i.e., BAliBASE and SABmark, the maximum number of sequences in a given file is only 28 and 23 respectively. To analyze the scalability of the proposed ProgSIO-MSA model we carried out numerous alignment experiments by varying the number of input sequences and recorded the running time (in milliseconds) of the proposed model. For this experiment, we considered the HomFam benchmark dataset Sievers *et al.* (2013). The number of input sequences in a query file is varied from 2 to 3000 and the respective running time of the proposed ProgSIO-MSA model is benchmarked and tabulated as shown in Table 3.9. Further, for the same set of input sequences running time of CLUSTAL Omega Sievers *et al.* (2011) is recorded and tabulated in Table 3.9.

The alignment running time of the ProgSIO-MSA and CLUSTAL Omega Sievers *et al.* (2011) are plotted in Figure 3.6. The X-axis indicates the number of sequences in a query file and the Y-axis represents the running time in milliseconds. Both X-

Table 3.9. Alignment Time(in milli secs) for ProgSIO-MSA and CLUSTAL Omega against the number of sequences of HomFam Test sets.

No. of Sequences	**ProgSIO-MSA in milli secs**	**CLUSTAL Omega in milli secs**
2	5	38
4	31	156
8	119	370
16	535	1122
25	1370	1883
50	5059	5826
100	21633	22281
200	64057	25127
250	98557	28096
500	423050	56392
1000	1778944	129608
2000	7156735	330917
3000	16104847	609076

axis and Y-axis values are on a logarithmic scale to avoid skewing towards the larger values. From the Figure 3.6, it can be observed that the proposed ProgSIO-MSA model is efficient in running time for up to 100 sequences; whereas, the CLUSTAL Omega is efficient for more than 100 sequences in a query file. This is mainly due to the following reasons: (i) the proposed ProgSIO-MSA model dynamically generates all possible pair-wise alignment every time after the best pair profile gets merged in the subsequence list (as explained in Steps 7 and 8 of Algorithm 3.3). By this dynamic approach, the proposed model is able to align more accurately with the trade-off of running time. (ii) the proposed model is implemented in single thread as there is less scope in parallelizing the model, i.e., (as mentioned above) due to the dynamic nature of the proposed model, it doesn't possess prior information of best pair to align parallelly. Thus, the proposed model is sequential and single-threaded; whereas the CLUSTAL Omega is featured with a multi-thread approach.

By this, we state that as the number of input sequences increases the alignment running time of the proposed model increases logarithmically.

D. Statistical Significance Analysis:

To analyze the statistical difference between the Q-Scores of the proposed ProgSIO-MSA model and each of the state-of-the-art models on the BAliBASE dataset, we performed the Wilcoxon Signed-Rank Test Corder and Foreman (2009), as the Q-Scores are not normally distributed. Let a null hypothesis indicate that there is no significant

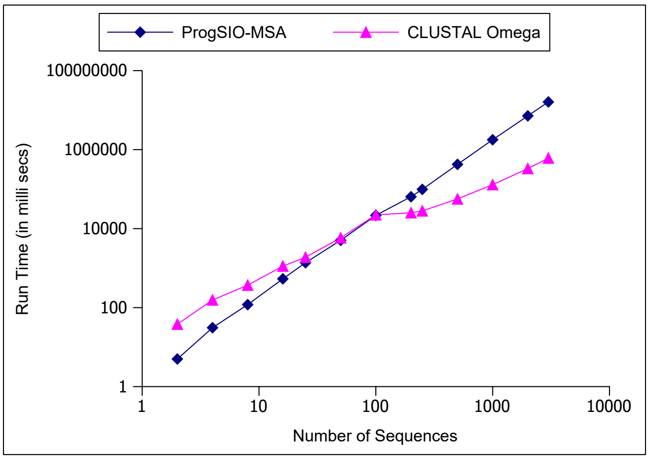


Figure 3.6. Scalability: Alignment time comparison of ProgSIO-MSA against CLUSTAL Omega for various number of sequences from HomFam test sets. Both axes are in logarithmic scales

difference between the proposed ProgSIO-MSA model and each of the state-of-the-art models for a significance level of 5% (i.e., 0.05). When $p \leq 0.05$, the hypothesis test for two sets of Q-Scores rejects this null hypothesis, which means that there is indeed a statistically significant difference between the same. Otherwise, i.e., when $p > 0.05$, the null hypothesis is retained and it indicates that there is no significant difference between the two sets of Q-Scores. The results of the Wilcoxon Signed-Rank Test are shown in Table 3.10 and the null hypothesis is rejected in all cases, except two. Hence, the accuracy of the proposed ProgSIO-MSA model is significantly higher than that of the progressive-based state-of-the-art models and is equally better with that of the one iterative model, i.e., GAPAM. Further, it is inferred that the proposed model, being a progressive approach to align multiple sequences with SIO, performs equally better when compared to GAPAM (best iterative model) and outperforms other iterative models, which runs for multiple iterations to achieve the same.

3.6 Summary

In this chapter, a more accurate and computationally feasible (polynomial time) alignment model was proposed with an effective scoring system and a novel optimization framework. The proposed scoring system incorporated two effective strategies, i.e., LBA and PRSDGP in which the LBA scoring strategy scores a current residue pair based on previous position status information and the PRSDGP scoring strategy dynamically calculates the gap penalty value based on its position and residue information using the mutation matrix. The proposed SIO framework identifies and optimizes the aligned results using the proposed scoring strategies to overcome the local optima limitation of the progressive approach. The proposed ProgSIO-MSA model being a progressive approach was evaluated against both progressive and iterative-based mod-

Table 3.10. Wilcoxon Signed-Rank Test between the Proposed ProgSIO-MSA Model and other State-of-the-art Models on BAliBASE Dataset

Approach	**Alignment Method**	**p-value ***	**Null Hypothesis Decision**	**Significant Difference (if p <0.05)**
Progressive	ML PIMA	<0.00001	Reject	Yes
	SB PIMA	<0.00001	Reject	Yes
	PILEUP8	<0.00001	Reject	Yes
	MULTALIGN	<0.00001	Reject	Yes
	CLUSTAL W/X	<0.00001	Reject	Yes
Stochastic	RBT-GA	<0.00001	Reject	Yes
	SAGA	0.00016	Reject	Yes
	MSA-GA	0.00236	Reject	Yes
	GAPAM	0.43540	**Retain**	**No**

* p-values are calculated up to five decimal point.

els on two benchmark datasets, i.e., BAliBASE and SABmark. The experimental results showed that the accuracy (quality) of the proposed ProgSIO-MSA model, when compared with state-of-the-art progressive models, was increased by at least 27.2% and 23% for Ref.2 and Ref.3 datasets of BAliBASE respectively. The quality of the proposed model outperformed state-of-the-art progressive-based iterative models for SABmark datasets. The proposed ProgSIO-MSA model performance is equally good when compared to the stochastic-based iterative model, i.e., GAPAM. Moreover, the computational efficiency of the proposed ProgSIO-MSA model outperformed CLUSTAL Omega in running time and outperformed GAPAM in time complexity by a factor of $[G\ .\ P]$ (for G number of generations and P number of populations). Further, the Q-Score differences between the proposed ProgSIO-MSA model and other state-of-the-art models were analyzed using a non-parametric statistical test with a significance level of 5% on BAliBASE datasets. Wilcoxon Signed-Rank Test results concluded that the quality of the proposed ProgSIO-MSA model significantly outperformed progressive-based state-of-the-art models and it is on par with the GAPAM. By experimental and statistical analysis, we conclude that the proposed ProgSIO-MSA model is a more accurate and computationally efficient model to perform multiple protein sequence alignment.

Chapter 4

Protein Secondary Structural Class Prediction

In this chapter[1], effective protein secondary structural class prediction models are proposed and discussed in detail. Two studies are carried out in solving PSSC prediction effectively. First, various sets of local features extracted from primary and secondary structural sequences are analyzed using state-of-the-art classifiers. Next, an effective set of global and local sets of features are extracted and these features are used to classify protein secondary structural classes based on the proposed ensemble of classifiers.

4.1 Preamble

Identification of Protein Secondary Structural Class (PSSC) information is one of the important activities in the analysis of protein structure and its functions. The Structural Classification of Proteins - Extended (SCOPe) is one of the largest publicly available protein databases in which proteins have been classified to determine the evolutionary relationship among proteins. The majority of proteins and their domains are manually curated of known structure in a hierarchy according to structural and evolutionary relationships. According to the latest extended version of the SCOPe 2.07 database[2], the proteins are majorly categorized into seven classes, namely, (1) All–α, (2) All–β, (3) α/β, (4) α+β, (5) Multi-domain proteins, (6) Membrane and cell surface proteins and (7) Small proteins. Over the years, it was observed that 90% of these protein entries consistently belong only to the first four structural classes Murzin *et al.* (1995), Andreeva *et al.* (2004), Andreeva *et al.* (2007). Therefore, this study concentrates on predicting the first four structural classes, i.e., All–α, All–β, α/β, and α+β.

The PSSC prediction is a multi-class classification problem in which an amino acid sequence is classified into any one of the four structural classes. This problem is addressed using machine learning techniques in which protein sequences are initially transformed into a fixed-size feature vector and later, the classifier model utilizes the feature vector to train itself to predict PSSC.

---

[1] The work described in this Chapter has been published in: **Sanjay Bankapur** and Nagamma Patil, ”Protein Secondary Structural Class Prediction Using Effective Feature Modeling and Machine Learning Techniques” in *IEEE 18th International Conference on Bioinformatics and Bioengineering (BIBE)*, Taichung, Taiwan. Oct-2018, IEEE.
Prince Kumar, **Sanjay Bankapur**, and Nagamma Patil, ”An enhanced protein secondary structure prediction using deep learning framework on hybrid profile based features” in *Applied Soft Computing, Elsevier*, 86, p.105926, 2020
Sanjay Bankapur and Nagamma Patil, ”Enhanced Protein Structural Class Prediction using Effective Feature Modeling and Ensemble of Classifiers” in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, (In Press, 2020).

[2] `http://scop.berkeley.edu/statistics/ver=2.07`

4.2 Datasets

To assess the performance of the proposed model against state-of-the-art models, we have considered five publicly available benchmark datasets.

Benchmark Datasets:
The first dataset consists of 277 protein sequences and the second dataset consists of 498 sequences which are constructed by Zhou Zhou (1998) and denoted as z277 and z498 respectively. Both the datasets, despite possessing high similarity (about 80%), are widely used to validate the prediction models. To analyze the performance impact of the proposed model on low similarity datasets, we have considered three other benchmark datasets. 25PDB Kedarisetti *et al.* (2006), 1189 Wang and Yuan (2000), and FC699 Kurgan *et al.* (2008) datasets which contain 1673, 1092, and 858 protein sequences respectively and exhibit less than 40% sequence similarity. All the protein sequences from these five benchmark datasets are span across the four classes of protein secondary structure such as All–α, All–β, α/β, and $\alpha+\beta$. The data characteristics and frequencies for each class are shown in Table 4.1.

Table 4.1. Data Characteristics of Five Benchmark Datasets

Dataset	**Sequence Similarity**	**Number of Protein Sequences**				
		All–α	**All–β**	α/β	$\alpha+\beta$	**Total**
z277 Zhou (1998)	High	70	61	81	65	277
z498 Zhou (1998)	High	107	126	136	129	498
25PDB Kedarisetti *et al.* (2006)	Low ($\leq$25%)	443	443	346	441	1673
1189 Wang and Yuan (2000)	Low ($\leq$40%)	223	294	334	241	1092
FC699 Kurgan *et al.* (2008)	Low ($\leq$40%)	130	269	377	82	858

4.2.1 Data Preparation

From a protein sequence, every amino acid residue can be predicted to one of the possible secondary structural elements such as Helix (H), Sheets (E), or Coil (C). By this, a secondary structural sequence can be generated from a query protein sequence.

For example: consider a sample protein sequence, say S_1 = GEYFTLQIRGRERFEMFRELNEALELKDAQA and its corresponding structural sequence, say $StrS_1$ is CCEEEEEECHHHHHHHHHHHHHHHCCCHHCC. The generated secondary structural sequence exhibit the same length as of protein sequence.

To generate a secondary structural sequence, many state-of-the-art models exist. In this study, we adopt two methods PSI-BLAST based secondary structure PREDiction

(PSIPRED) method McGuffin *et al.* (2000) and (ii) Hybrid model on evolutionary-based profiles using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) Kumar *et al.* (2020).

PSIPRED: this method incorporates two sequential feed-forward neural networks to predict the structural elements with the help of Position-Specific Iterative Basic Local Alignment Search Tool (PSI-BLAST) profiles Jones (1999). PSI-BLAST Altschul *et al.* (1997) is an alignment search tool to generate Position-Specific Scoring Matrix (PSSM) profiles. PSI-BLAST takes a query protein sequence as input and compares it to a protein database to shortlist similar protein sequences and then multiple sequence alignment is performed on shortlisted sequences to generate PSSM profiles. For a query protein sequence (say, S), the protein secondary structural sequence generated from PSIPRED is denoted as $StrS^P$.

Hybrid CNN+RNN: this method extracts features from two evolutionary profiles such as Position-Specific Scoring Matrix (PSSM) and hidden Markov model (HMM) profiles and secondary structural sequence is predicted using the combination of CNN followed by RNN framework.

PSSM: The PSSM profiles are generated using PSI-BLAST Altschul *et al.* (1997) tool by searching a protein sequence on National Center for Biotechnology Information (NCBI's) non-redundant (NR) database with 0.001 as a cut-off value over three iterations. The generated PSSM profile contains a linear substitution probability matrix and a log-odd matrix. Both the matrices contain 20 columns where each column indicates an amino acid. In this study, the linear substitution probability matrix of the PSSM is considered for the feature extraction as they exhibit rich information of amino acid possible substitutions.

HMM: The HMM profiles are generated using HHblits Remmert *et al.* (2012) tool in which a protein sequence was searched on the Uniprot20 database with 0.001 as a cut-off value over four iterations. The generated HMM profile contains a matrix of 30 columns in which the first 20 columns indicate the position-specific probabilities of 20 amino acids. The next ten columns indicate the probabilities of three states such as insertion, deletion, and match, which are defined in HHblits. The first 20 columns of HMM profiles are considered for the feature extraction as they exhibit rich information of amino acid possible position-specific probabilities.

For a query protein sequence (say, S), the protein secondary structural sequence generated from Hybrid CNN+RNN is denoted as $StrS^H$.

4.3 PSSC Prediction using Local Features with State-of-the-art Classifiers

A preliminary study is carried out in order to identify the best performing feature sets and classifiers.

4.3.1 Datasets

In this preliminary study, three benchmark datasets namely z277, 25PDB, and FC699 are considered. More details on these three datasets are available in section 4.2.

4.3.2 Feature Extraction

Quality features play a major role in predicting PSSC accurately. Two novel feature extraction techniques namely SkipXGram bi-gram (SXGbg) and Character Embedding (CE) are proposed to extract local amino acid interactions (local features).

4.3.2.1 SkipXGram bi-gram (SXGbg) Technique

Protein secondary structure is mainly due to the hydrogen bonds among two amino acid molecules. Hence the proposed model concentrates and extracts bi-gram features. Moreover, one turn of α-helix is observed to be exhibited on an average of 3.6 amino acid residues Segrest *et al.* (1999), therefore to mimic the α-helix nature, the proposed SXGbg technique extracts all possible bi-grams by skipping X-grams between the two residues, where, X value varies from 0 to 5 in our experiment. By this, six sets of bi-gram features are generated from protein sequences where each set consists of 400 features. Six more sets of bi-gram features are extracted from secondary structural sequences in which each set consists of 9 features, as a secondary structural sequence is represented using three elements - H (Helix), E (Sheets), and C (Coil). Let S be a protein sequence, which is made up of amino acid residues, of length L i.e. r_0, r_1 .. , r_{L-1} where r_0 is the residue at first position and r_{L-1} is the residue at L^{th} position. From the protein sequence S, the bi-gram features are extracted and added to the SXGbg feature set and it is as shown in equation 4.1.

$$SXGbg(S) = \sum_{i=X}^{L-1} r_{(i-X)}.r_{(i+1)} \tag{4.1}$$

Where, X indicates the number of skipped grams and the values are varied from 0 to 5 to obtain six set of SXGbg features.

Procedure to generate SXGbg set of features is shown in Algorithm 4.1. The Algorithm 4.1 takes n protein sequences (from a dataset) and a skip value, i.e., X as inputs to produce an output of size n x 400. Where n represents the number of protein sequences and 400 represents the feature vector (as there are 20 amino acids, 20 x 20 = 400) for each protein sequence.

Algorithm 4.1. : Proposed Algorithm to Extract SkipXGram bi-gram Features

Input: List of n protein sequences of variable length and a SkipGram value, X

Output: SXGbg feature set with the occurrence count for all n protein sequences

```
for each protein sequence j do
    $L$ be a $j^{th}$ sequence length
    for i=leave to $L$ do
        bg=char at [i-X] and [i+1] from the $j^{th}$ sequence
        if($SXGbg_j$ in [bg]) then
            Increment the bg count by 1 in $SXGbg_j$ set
        else
            Add the bg to $SXGbg_j$ set with count=1
    end for
end for
```

Table 4.2 lists all the six sets of features for varying X values which are extracted from the proposed SkipXGram algorithm for a sample protein sequence say S= KLMTPTRS. For the S0Gbg feature set, seven features that are mentioned in Table 4.2 exhibit a count value 1 as their frequency occurred only one time and the rest of the features (i.e., 400-7=393) are set to value 0 as they didn't appear in the sample sequence. Similarly, S1Gbg to S5Gbg are calculated. If any multiple occurrences of a bi-gram feature then its count is incremented respectively and it is shown in Step 6 of the proposed Algorithm 4.1.

Table 4.2. Six Sets of Features from Proposed SXGbg Technique for a Sample Protein Sequence, S = KLMTPTRS

X Value	**Feature Set Name**	**Features**
0	S0Gbg	{KL, LM, MT, TP, PT, TR, RS}
1	S1Gbg	{KM, LT, MP, TT, PR, TS}
2	S2Gbg	{KT, LP, MT, TR, PS}
3	S3Gbg	{KP, LT, MR, TS}
4	S4Gbg	{KT, LR, MS}
5	S5Gbg	{KR, LS}

4.3.2.2 Character Embedding (CE) Technique

We adopted and modified the Word2Vec word embeddings technique Mikolov *et al.* (2013) such that it generates character embeddings, where each character is a protein residue. The embedding model represents each protein sequence into a vector of size

400. As similar to the ability of the Word2Vec model to map words belonging to the same domain in close proximity in the vector space, the character embedding model works in such a way that the residues which share similar characteristics in protein sequence are placed in close vicinity in the vector space. For a query protein sequence, this technique is applied on both primary and secondary structural sequences to extract a fixed feature vector of size 400 each.

4.3.3 Classification

The protein secondary structural class prediction is a multi-class classification problem. The quality features are extracted using the proposed feature extraction techniques and from these extracted features, classification is performed using various machine learning techniques. In this preliminary study, we have considered the most popular state-of-the-art machine learning classification techniques such as logistic regression, k nearest neighbor classifier, multi-layer perceptron, support vector machine, gradient boosting machines, and random forest to classify the given protein sequences into its respective secondary classes.

- Logistic Regression (LR): It is a logit model used to predict the log-odds probability of a dependent variable or a prediction response based on linear combinations of one or more independent variables or features. In this study, the multinomial logistic regression method is implemented which uses L2 regularization to calculate the loss and maximum likelihood estimation to predict the probability of category membership on four possible secondary structural class outcomes.

- K-Nearest Neighbour (k-NN): k-NN is an instance-based machine learning technique in which all the available instances are stored, and for the new instance, the distance measure is evaluated among the stored k nearest neighbor instances and is assigned to the respective class using majority voting. k-NN classifier works accurately only if the training instances are linearly separable with higher-margin among the class labels. It was observed $k = 4$ performed the best in the Benchmark dataset.

- Multi-Layer Perceptron (MLP): MLP is one of the popular classifiers which is based on a feed-forward artificial neural network. An MLP exhibits a minimum of three layers of processing nodes, and except the input layer nodes, all the other nodes utilize a nonlinear activation function. MLP uses a gradient descent (back-propagation) approach to minimize the prediction error function of the interconnected network with the help of the training dataset. In this study, we implemented this method with three hidden layers consisting of 100 processing nodes at each layer and five output nodes at the output layer to predict sub-chloroplast locations. For all the nodes at hidden layers and output layer, we adopted the $tanh$ activation function due to its effective adaptability towards nonlinear datasets.

- Support Vector Machine (SVM): SVM is a supervised machine learning model. It is one of the widely used supervised learning models since it performs effectively in tackling problems across various domains. SVM trains the model in such a way that it finds the hyperplane that maximizes the margin among the given classes using kernel function, thereby minimizing the prediction error. This method is useful in solving pattern recognition problems. In this study, we implemented the SVM method with Radial Basis Function (RBF) as the kernel, since RBF outperforms both linear ad polynomial kernels due to its tolerance to input noise with generalization ability.

- Gradient Boosting Machine (GBM): GBM is an ensemble classifier in which multiple weak predictors are made to learn in a sequence to minimize the loss function. The main aim of this method is to boost the prediction accuracy by ensembling weaker predictors in a sequence such that subsequent predictors learn from the mistakes of previous predictors. In this study, we have implemented this method in which regression decision trees are considered as weak predictors and negative gradient multinomial deviance as the loss function.

- Random Forest (RF): It is one of the most effective ensemble classifiers in which multiple decision trees are generated during the training phase, which are allowed to split randomly from a seed point. These randomly generated decision trees are collectively termed as 'forest.' Each decision tree, as the result of this model, depicts the given dataset into a tree structure where the root node constitutes the most discriminating feature/attribute, whereas the leaf node maps to the prediction class label. It is evident that the higher the number of decision trees, the better the prediction accuracy. Hence, this method is implemented with the number of decision trees equal to 360, which was identified empirically, and it was observed that with further increase in the trees, there was no further improvement in the prediction accuracy

4.3.4 Proposed Model

The proposed preliminary model to predict PSSC is as shown in Figure 4.1. In this study, a better performing data preparation approach, better-performing feature sets, and better classifier are shortlisted. The detailed experiment analysis is elaborated in the next section 4.3.5

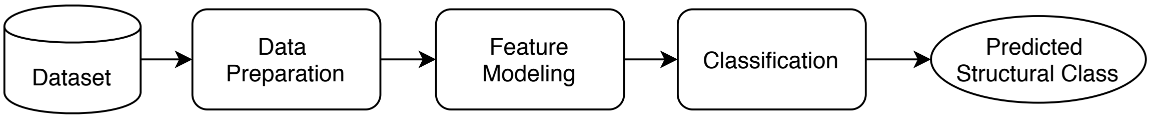


Figure 4.1. The Proposed Model for the PSSC Prediction based on Local Features.

4.3.5 Results and Discussion

In this section, the performance analysis of CE and SXGbg features are analyzed in detail. The best performing classifier is shortlisted using the using effective sets of features. All the experiment in this study is carried out using ten-fold cross-validation.

4.3.5.1 Environment Setup and Performance Measure

The SXGbg feature extraction technique has been implemented in Java, Eclipse Platform 3.8.1. The CE feature extraction technique has been implemented in Python. The state-of-the-art classifiers are implemented in Python using Scikit-learn Pedregosa *et al.* (2011).

The performance of the proposed model was evaluated using the Overall Accuracy metric. Overall Accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases examined and it is shown in equation 4.2

$$OverallAccuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.2}$$

Where TP, FP, TN, and FN are the total number of true positives, false positives, true negatives, and false negatives respectively.

4.3.5.2 Performance Analysis on Character Embedding Features

In this analysis, the effectiveness of CE features from PSIPRED and Hybrid CNN+RNN is carried out. To perform this analysis, two categorizations are made: (i) CE-PSIPRED - in which two sets of CE-based features are extracted from protein sequence (S) and secondary structural sequence ($StrS^P$) respectively. Each feature set is of size 400 and the CE-PSIPRED feature vector constitutes to of size 800. (ii) CE-Hybrid CNN+RNN - in which two sets of CE-based features are extracted from protein sequence (S) and secondary structural sequence ($StrS^H$) respectively. Each feature set is of size 400 and the CE-Hybrid CNN+RNN feature vector constitutes to of size 800.

The effectiveness of these two CE-based feature categories is carried out using various state-of-the-art classifiers (that were discussed in section 4.3.3). The PSSC prediction results of each of the classifiers are tabulated in Table 4.3.

Discussion: From the Table 4.3, it can be observed that the RF classifier outperformed other classifiers. Further, it is also observed that the structural class predic-

Table 4.3. The Performance Comparison (in percentage) of Structural Class Prediction based on the CE Features Extracted from the PSIPRED against Hybrid CNN+RNN

Classifiers	**z277**		**25PDB**		**FC699**	
	CE-PSIPRED	**CE-Hybrid CNN+BRNN**	**CE-PSIPRED**	**CE-Hybrid CNN+BRNN**	**CE-PSIPRED**	**CE-Hybrid CNN+BRNN**
LR	65.7	73.3	72.1	72.4	83.5	82.9
k-NN	81.4	84.2	72.3	**72.7**	83.7	84.5
MLP	75.2	78.5	73.5	72.8	83.7	85.2
SVM	81.0	80.2	74.4	73.9	86.2	86.5
GBM	86.2	86.6	73.7	73.5	86.1	85.7
RF	86.8	**87.4**	**74.8**	73.9	**87.7**	86.1

tion accuracy of CE-Hybrid CNN+RNN features is higher than that of CE-PSIPRED features for high similarity dataset i.e., z277. However, CE-PSIPRED features outperformed CE-Hybrid CNN+RNN features for low similarity datasets, i.e., for 25PDB and FC699. This is due to the fact that PSIPRED generates secondary structure using the vast knowledge of higher protein domains from the Non-Redundant (NR) database; whereas, Hybrid CNN+RNN generates secondary structural sequences using limited knowledge of protein sequences from the CB6133 database. Hence, PSIPRED is effective in low-similarity datasets. Along with Random Forest (RF) classifier, two feature sets of CE-PSIPRED namely CE-Seq (extracted from protein sequences) and CE-Str (extracted from PSIPRED secondary structural sequences) are shortlisted for further analysis.

4.3.5.3 Performance Analysis on SkipXGram bigram (SXGbg) Features

As discussed in the section 4.3.2.1, six sets of features (by varying X value from 0 to 5) are extracted on protein sequences using the proposed SXGbg technique and each set constituting 400 features. Using the same technique, six more sets (by varying X value from 0 to 5) of features are extracted from PSIPRED-based secondary structural sequences also, in which each set constitutes to 9 features, since the structural sequence is made up of three characters indicating the amino acid residue is either Helix (H), Sheet (E) or Coil (C). Therefore, a total of twelve sets of features are extracted for a given dataset using the SXGbg technique. Let *SXGbg-Seq* be the feature sets extracted from protein sequences and *SXGbg-Str* be the feature sets extracted from PSIPRED-based secondary structural sequences. These twelve sets of features are evaluated on one high-similarity dataset (z277) and two low-similarity benchmark datasets (i.e. 25PDB and FC699). The PSSC prediction performance of SXGbg features using Random Forest (RF) classifier are shown in Table 4.4.

From Table 4.4, it is evident that the prediction accuracy improves for the structural

Table 4.4. The Performance Comparison of PSSC Prediction using SXGbg Features and Random Forest Classifier

Datasets	**z277**		**25PDB**		**FC699**	
SXGbg Features	**Protein Sequence**	**Structural Sequence**	**Protein Sequence**	**Structural Sequence**	**Protein Sequence**	**Structural Sequence**
S0Gbg	77.78	83.75	52.97	76.05	68.30	88.32
S1Gbg	76.34	83.74	53.49	75.34	68.53	88.55
S2Gbg	78.18	83.74	53.01	75.88	67.95	88.44
S3Gbg	**79.87**	**85.14**	**53.67**	**77.19**	**68.76**	**89.72**
S4Gbg	78.17	81.93	51.52	76.77	68.30	87.97
S5Gbg	78.24	81.94	50.87	77.10	67.12	88.67

sequence when compared to the protein sequence for all three datasets. From this, we can conclude that the information present in structural sequences is much higher than in protein sequences. Moreover, S3Gbg sets of features (i.e. S3Gbg-Seq & S3Gbg-Str) which are extracted from protein sequence and structural sequence reported high prediction accuracy consistently across all the three datasets when compared to other sets of features. Hence, these two feature sets (i.e. S3Gbg-Seq & S3Gbg-Str) extracted from the SXGbg technique are shortlisted and considered for further analysis.

4.3.5.4 Performance Analysis on the Proposed Feature Model using State-of-the-art Classifiers

From the above discussed sections 4.3.5.2 and 4.3.5.3, four effective feature sets are shortlisted from CE and SXGbg techniques. All the four shortlisted sets of features (a total of 1209 features) are further combined and evaluated using the state-of-the-art classifiers. The performance comparison using various classifiers on both datasets is as shown in Table 4.5. The combination of the proposed feature modeling with RF classifier reported better prediction accuracy consistently for all the three datasets and the same is evident from the Table 4.5 and hence, this combination (CE-Seq + CE-Str + S4Gbg-Seq + S4Gbg-Str + RF) is considered as the proposed model of this preliminary study.

4.3.5.5 Performance Analysis of the Proposed Model against State-of-the-art Models

In this preliminary study, the proposed model consists of effective feature sets (four sets of features) and RF as the classifier. The performance of the proposed model is evaluated against state-of-the-art models on three benchmark datasets and the respective results are shown in the Tables 4.6, 4.7, and 4.8.

Table 4.5. The PSSC Prediction Evaluation (in %) of State-of-the-art Classifiers' using the Proposed Feature Modeling on the Three Benchmark Datasets

Classifier	**z277**	**25PDB**	**FC699**
	CE-SXG Features	**CE-SXG Features**	**CE-SXG Features**
LR	85.73	76.14	86.94
k-NN	86.73	74.24	87.99
MLP	85.41	76.81	89.86
SVM	87.04	77.11	90.21
GBM	86.47	77.59	90.09
RF	**87.78**	**79.79**	**91.61**

Table 4.6. The Performance Comparison (in percentage) of the Proposed Model against State-of-the-art Methods for z277 Dataset

Models	**All–α**	**All–β**	α/β	$\alpha+\beta$	**Overall Accuracy**
IGA-SVM Li *et al.* (2008)	84.30	85.50	92.60	70.70	84.50
CWT-PCA-SVM Li *et al.* (2009)	85.70	90.20	87.70	80.10	85.90
Information Theoretical Zheng *et al.* (2010)	87.10	80.30	93.80	67.70	83.00
NN-CDM Liu *et al.* (2010*b*)	80.00	86.40	91.60	81.80	85.20
LZ-BMKL Mao *et al.* (2013)	92.90	85.30	92.60	69.20	85.60
Dehzangi et al. Dehzangi *et al.* (2013*b*)	90.00	93.40	80.00	96.30	**90.30**
PMCI-RFE Yuan *et al.* (2018)	-	-	-	-	84.43
Proposed Model (this study)	91.29	92.09	93.30	84.77	87.78

From the Tables 4.7 and 4.8, it is evident that the performance of the proposed model outperforms other state-of-the-art models by a factor of 3% to 23% and 4% to 6% on 25PDB and FC699 datasets respectively. From Table 4.6, it can be observed that the proposed model on the z277 dataset was effective when compared to other models except Dehzangi *et al.* (2013*b*) model. This is due to the fact that Dehzangi *et al.* (2013*b*) model was ensembled with different classifiers to solve PSSC prediction.

The overall outcome of this preliminary study is listed below:

- Features from protein secondary structural sequences are more effective than protein sequences.
- The prediction performance of CE-PSIPRED based features is higher when compared to CE-Hybrid CNN+RNN features.
- S3Gbg feature set outperformed in comparison to other SXGbg features for all

Table 4.7. The Performance Comparison (in percentage) of the Proposed Model against State-of-the-art Methods for 25PDB Dataset

Models	**All–α**	**All–β**	α/β	$\alpha+\beta$	**Overall Accuracy**
Stacking Ensemble Kedarisetti *et al.* (2006)	-	-	-	-	59.90
LLSC-PRED Kurgan and Chen (2007)	75.20	67.50	62.10	44.00	62.20
AAD-CGR Yang *et al.* (2009)	64.30	65.00	65.00	61.70	64.00
AADP-PSSM Liu and Jia (2010)	83.30	78.10	76.30	54.40	72.90
AAC-PSSM-AC Liu *et al.* (2012)	85.20	81.30	73.70	55.20	73.90
Ensemble Model Dehzangi *et al.* (2013*b*)	86.10	80.80	80.60	60.10	76.70
Proposed Model (this study)	92.70	78.90	71.90	74.50	**79.79**

Table 4.8. The Performance Comparison (in percentage) of the Proposed Model against State-of-the-art Methods for FC699 Dataset

Models	**All–α**	**All–β**	α/β	$\alpha+\beta$	**Overall Accuracy**
SCPRED Kurgan *et al.* (2008)	-	-	-	-	87.50
CBF-PSSE Dai *et al.* (2013)	84.62	91.45	93.90	34.50	86.01
PBF-PSSE Dai *et al.* (2013)	88.46	81.41	88.86	80.49	85.66
Proposed Model (this study)	96.40	92.50	95.10	65.10	**91.61**

the three benchmark datasets.

- Combination of CE and S3Gbg features from protein sequences and secondary structural sequence enhanced overall prediction performance.

- RF classifier consistently outperformed other state-of-the-art classifiers.

- The proposed PSSC prediction model outperformed other state-of-the-art PSSC prediction models for low-similarity datasets.

- The proposed PSSC prediction model was less effective by a margin of 2.5% than Dehzangi *et al.* (2013*b*) model on a high-similarity dataset.

The performance of the proposed model of this preliminary study is further improved by enhancing feature modeling and with the ensemble of classifiers. This will be discussed in the next section 4.4 in detail.

4.4 PSSC Prediction using Local and Global Features with Ensemble of Classifiers

The previously discussed PSSC prediction model is further enhanced and analyzed in this section.

4.4.1 Datasets

In this enhanced study, the proposed model is evaluated on benchmark datasets as well as on the derived latest dataset.

A. Benchmark Datasets:

The proposed model is evaluated on five publicly available benchmark datasets that had been discussed in the previous section 4.2.

B. Latest Low-similarity High-volume Dataset: To validate the performance consistency and robustness of the proposed model, we have derived a dataset consisting of a high-volume of newly discovered protein sequences based on two aspects, such as (1) protein sequences that are extracted from the latest extended version of SCOP i.e. SCOPe 2.07 database Fox *et al.* (2013), and (2) all the protein sequences that exhibit $\leq$25% sequence similarity. Henceforth, this dataset will be referred to as SCOPe_2.07. The data characteristics and number of protein sequences for each structural class are shown in Table 4.9.

Table 4.9. Data Characteristics of Latest Large-Scale Low-similarity Dataset

Dataset	**SCOPe Version**	**Sequence Similarity**	**Number of Protein Sequences**				
			All–α	**All–β**	α/β	α+β	**Total**
SCOPe_2.07	2.07-Stable (March 2018)	Low $\leq$25%	1760	1791	2174	2181	7906

4.4.2 Data Preparations

In this study, only PSIPRED McGuffin *et al.* (2000) method is adopted to generate protein secondary structural sequences from protein sequences. For all the datasets, we prepared the structural sequence for every input protein sequence.

4.4.3 Feature Modeling

Feature modeling is an important step to extract features by transforming raw protein sequences into feature vectors of a fixed-size which exhibit discriminating information in predicting the PSSC accurately. In this study, we propose an enhanced feature modeling approach to extract the global and local discriminating features from the amino acid sequences and generated structural sequences. The local-based features are extracted using two proposed techniques as Character Embedding (CE) and SkipXGram bi-gram (SXGbg). The global-based features are extracted using General Statistical (GS) technique.

4.4.3.1 Character Embedding (CE) Technique

Character Embedding technique follows an unsupervised learning approach to train and generate the vector space. Word embedding is a word vectorization technique that transforms a word into a contiguous vector such that similar words are mapped in the vicinity in the vector space and the generated vectors are dense, real-values with limited lower dimensions.

In our earlier investigation (section 4.3.5.2) to predict the protein structural classes using the Word2Vec skip-gram architecture reported a satisfactory result. In this study, we considered three popular word embedding models such as Word2Vec Mikolov *et al.* (2013), GloVe Pennington *et al.* (2014), and fastText Joulin *et al.* (2016) and modified these models such that they return character embeddings, where each character is a residue of a protein sequence.

Word2Vec Mikolov *et al.* (2013) and fastText Joulin *et al.* (2016) are predictive models where each model exhibits two architectures, namely, Contiguous Bag-Of-Words (CBOW) and Skip-Gram (SG). The training phase of CBOW architecture predicts the current word from a window of context words. In contrast, SG architecture predicts the window of context words from a current word. GloVe (Global Vectors) Pennington *et al.* (2014) is a statistical count-based model in which the model learns its vectors by training on non-zero entries in a word-word co-occurrence matrix. Both Word2Vec and GloVe models train the network by treating each word from the corpus as an atomic entity, whereas, fastText model trains by treating each word as a set of characters.

To extract embedding-based feature sets, we have explored a total of five embedding architectures, such as Word2Vec Contiguous Bag-Of-Words (W2V-CBOW), Word2Vec Skip-Gram (W2V-SG), GloVe, fastText Contiguous Bag-Of-Words (fastText-CBOW), and fastText Skip-Gram (fastText-SG).

Two sets of features are extracted using the character embedding approach from a query protein sequence and its respective secondary structural sequence — each feature set consisting of 400 features.

The secondary structural sequence is further processed to generate a structural sequence code by removing the Coil elements (i.e., C, as coil doesn't contribute in the prediction of structural class) and by replacing the contiguous repetition of the same structural element with the combination of a total number of occurrences and its structural element. For example, the structural sequence, say $StrS_1$ = CCEEEEEECHHHHHHHHHHHHHHHCCCHHCC after removing Coil elements will be EEEEEEHHHHHHHHHHHHHHHHH and processed to generate structural sequence code, $StrSCode_1$ = 6:E–17:H, where the contiguous repetition of segment E (i.e., EEEEEE) is replaced by its number of occurrences and the structural element (i.e., 6: E), and similar activity is performed for rest of the sequence. From the resulting structural sequence code (StrSCode) information, one more set of 400 features is extracted using the word embedding approach. The contiguous frequency and its structural element separated by a colon constitute a word, i.e., 6:E and 17:H are the two words of $StrSCode_1$. Using the embedding technique, a total of three sets, each consisting of 400 features is extracted, making it a 1200 feature vector. The 1200 embedding-based feature vector represents an effective spatial arrangement of amino acid sequences, secondary structural sequences, and structural sequence codes.

4.4.3.2 SkipXGram bi-gram (SXGbg) Technique

The most common types of protein secondary structures are the α-helices (H) and the β-sheets (E). Both these structures are formed due to the hydrogen bond between two residues. Moreover, there are 3.6 residues per turn in an α-helix structure. To mimic these biological characteristics, we have extracted various skipped bi-gram feature sets by adopting SkipXGram Technique (SXG). Using the SXG technique, we have extracted six sets of skipped bi-gram features from protein sequence as well as secondary structural sequence, and each set consisting of 400 feature size.

4.4.3.3 General Statistical (GS) based Feature

Along with the features extracted using E and SXGbg techniques, a set of 9 general statistical (GS) based features are generated to cover the global information of a structural sequence and those are:

- $f_H(StrS_{H,E,C})$: The frequency of an element H from a structural sequence with respect to the length of $StrS_{H,E,C}$.
- $f_E(StrS_{H,E,C})$: The frequency of an element E from a structural sequence with respect to the length of $StrS_{H,E,C}$.
- $f_C(StrS_{H,E,C})$: The frequency of an element C from a structural sequence with respect to the length of $StrS_{H,E,C}$.
- $Max_H(StrS_{H,E,C})$: Ratio contributing to the maximum number of consecutive

H elements in a structural sequence.

- $Max_E(StrS_{H,E,C})$: Ratio contributing to the maximum number of consecutive E elements in a structural sequence.
- $Max_C(StrS_{H,E,C})$: Ratio contributing to the maximum number of consecutive C elements in a structural sequence.
- $f_H(StrS_{H,E})$: The frequency of an element H from a structural sequence without the C elements with respect to the length of $StrS_{H,E}$.
- $f_E(StrS_{H,E})$: The frequency of an element E from a structural sequence without the C elements with respect to the length of $StrS_{H,E}$.
- $Length(StrSCode_{H,E})$: The total length of a structural sequence code.

Let $StrS_{H,E,C}$=CCEEEEEECHHHHHHHHHHHHHHHCCCHHCC be a sample secondary structural sequence. The total number of H, E, C elements are 17, 6, 8 respectively. The length of the $StrS_{H,E,C}$ is 31. Maximum continuous occurrences of H, E, C elements are 15, 6, 3 respectively. After removing the C elements from the secondary structural sequence, i.e., $StrS_{H,E}$=EEEEEEHHHHHHHHHHHHHHHHH. The length of the $StrS_{H,E}$ is 23. The structural sequence code of $StrS_{H,E}$ will be 6:E-17:H and its length is 8. The above mentioned 9 feature values are as follows: $f_H(StrS_{H,E,C})$ = 17/31, $f_E(StrS_{H,E,C})$ = 6/31, $f_C(StrS_{H,E,C})$ = 8/31, $Max_H(StrS_{H,E,C})$ = 15/31, $Max_E(StrS_{H,E,C})$ = 6/31, $Max_C(StrS_{H,E,C})$ = 3/21, $f_H(StrS_{H,E})$ = 17/23, $f_E(StrS_{H,E})$ = 6/23, and $Length(StrSCode_{H,E})$ = 8.

4.4.4 Classification

The prediction of PSSC is a multiclass classification problem. The relevant sets of features are extracted using the proposed enhanced feature modeling approach, and the extracted feature vectors are fed to a classification model as an input, to predict the PSSC. The majority of the existing works on the PSSC problem were carried out using a SVM classifier and have reported satisfactory prediction accuracy Ding *et al.* (2014); Qin *et al.* (2015).

In the literature, an ensemble of different classifiers are explored to address various challenges of protein sequence analysis Dehzangi *et al.* (2009, 2010) and the recent work Dehzangi *et al.* (2013*b*) has shown that the ensemble of classifiers is effective in addressing the PSSC problem. An ensemble of different classifiers facilitates prediction by combining the opinions of all classifiers via majority or probability-based voting. By this, the limitations of a classifier can be overcome with the strength of the other classifiers. Therefore, we explored various state-of-the-art classifier methods on all the benchmark datasets and proposed a generalized prediction model. The proposed generalized model works for all the categories of sequence similarity datasets (i.e. high to low), by an ensemble of three classifiers in parallel such as SVM, RF (bagging), and GBM (boosting) to work as single classification model.

Based on the preliminary investigation of various state-of-the-art classifiers, an ensemble of classifiers is proposed. The predicted output of the proposed ensemble classifier is based on the highest probability-based voting, i.e., each classifier outputs four probability values (for four classes) for a given protein sequence. Output probabilities of each class across the three classifiers are averaged and the query protein sequence is classified to the highest average probability class.

In this study, we implemented the SVM classifier with the penalty parameter C=4.0 and the Radial Basis Function (RBF) as the kernel function, since RBF relatively outperforms linear, polynomial, and sigmoid kernels due to its tolerance to input noise with generalizability. In the RF classifier, it is well known that the higher the number of decision trees, the lower the risk of the model being subjected to over-fitting, and the better the prediction accuracy. Hence, an RF classifier is implemented with the number of decision trees equal to 350, a value that is identified empirically. In our experiments, it was observed that any further increase in the number of trees, did not improve the prediction accuracy. We implemented the GBM classifier by choosing regression trees as weak predictors and negative gradient multinomial deviance as the loss function. Higher the number of weak predictors (boosting stages), gradient boosting is fairly robust to over-fitting. Therefore, the number of boosting stages is set to 350. It is worth noting that in all our experiments, the hyperparameters of the proposed ensemble classifier are constant, and then the prediction accuracies are recorded across all the datasets.

4.4.5 Proposed Model

The proposed ensemble classifier is trained on the sets of features that are extracted and shortlisted using the proposed feature modeling for PSSC prediction. In this study, a combination of the proposed feature modeling and the proposed ensemble classifier constitute the proposed model. The overall framework of the proposed model is as shown in Figure 4.2.

4.4.6 Results and Discussion

The performances of Embedding and SkipXGram based features are analyzed separately. The results of the best performing sets of features with the proposed ensemble

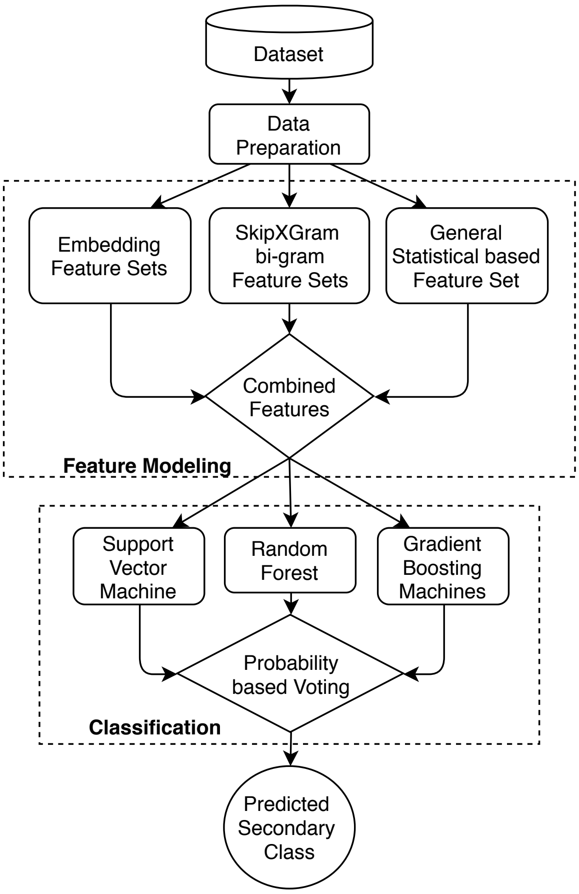


Figure 4.2. Framework of the proposed model constituting local and global features with ensemble of classifiers.

of classifiers on five benchmark datasets, i.e., z277, z498, 25PDB, 1189, and FC699 are analyzed. The overall prediction performances of the proposed model on benchmark datasets are compared with the state-of-the-art methods. Further, the prediction performance of the proposed model is validated on the large-scale updated dataset, i.e., SCOPe_2.07 which consists of a high volume of newly discovered protein sequences exhibiting $\leq 25\%$ sequence similarity.

Most of the published works on the PSSC problem are validated using the Jackknife approach Liu *et al.* (2012); Ding *et al.* (2014); Zhang *et al.* (2014). It was observed that the cross-validation evaluation approach produces results similar to Jackknife Efron and Gong (1983) and recent work on PSSC Zhang *et al.* (2014) showed that the overall prediction accuracy using 10-fold cross-validation on most of the datasets is slightly lesser when compared to Jackknife. Therefore, in this study, we have recorded all the performance measures using a 10-fold cross-validation approach. Detailed discussion on the performance analysis of various feature sets are as follows:

4.4.6.1 Experimental Setup and Performance Measures

The SXG and GS feature extraction techniques are implemented in Java, Eclipse Platform 3.8.1. The embedding approaches, such as the Word2Vec, GloVe, and fastText are implemented in Python. The proposed ensemble classifier is implemented in Python using Scikit-learn Pedregosa *et al.* (2011).

The performance of the proposed model was evaluated and benchmarked using four standard metrics such as Sensitivity (Sens), Specificity (Spec), Matthews correlation coefficient (MCC), and Overall Accuracy as given in equations 4.3, 4.4, 4.5, and 4.6 respectively. Sensitivity and specificity measure the proportion of actual positives and actual negatives that are correctly identified; Overall Accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases examined. These three metrics are measured in percentage. MCC takes into account of true positives, false positives, true negatives, and false negatives and is generally regarded as a balanced measure which can be used even if the classes are of varying sizes. The MCC value ranges from -1 to 1, where 0 indicates random correlation, the higher negative value (i.e., -1) indicates poor prediction quality, and a higher positive value (i.e., +1) indicates better prediction quality. These evaluation metrics are mathematically defined as follows:

$$Sens = \frac{TP}{TP+FN} \tag{4.3}$$

$$Spec = \frac{TN}{FP+TN} \tag{4.4}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{4.5}$$

$$OverallAccuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{4.6}$$

Where TP, FP, TN, and TN are the total number of true positives, false positives, true negatives, and false negatives respectively.

4.4.6.2 Performance Analysis of Embedding based Feature Sets

To effectively extract embedding-based features, we explored a total of five embedding architectures, such as W2V-CBOW, W2V-SG, GloVe, fastText-CBOW, and fastText-SG. From each architecture, three sets of features are extracted in which each set consists of 400 features. These three sets of features are extracted from protein sequences, structural sequences, and structural sequence codes which constitute 1200 features in total.

Table 4.10. The Performance Comparison (in percentage) of various Embedding Architectures on the Benchmark Datasets

Dataset	**Character Embedding Architectures**				
	W2V-CBOW	**W2V-SG**	**GloVe**	**fastText-CBOW**	**fastText-SG**
z277	86.00	87.80	86.00	**91.10**	88.54
z498	96.70	**96.76**	95.95	96.55	94.54
25PDB	75.50	74.64	72.84	**75.58**	72.73
1189	**77.01**	76.62	76.17	76.61	72.87
FC699	88.90	89.03	86.52	**89.05**	89.04
SCOPe_2.07	74.30	74.52	73.90	**74.58**	73.57

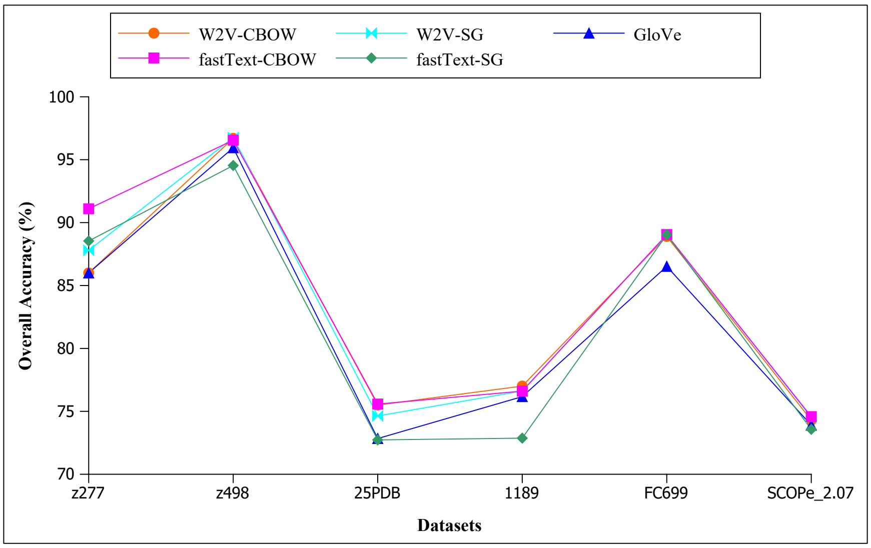


Figure 4.3. The Performance Variations of Embedding Architectures on Benchmark Datasets.

The overall performance comparisons of all the five embedding architectures across six benchmark datasets are tabulated in Table 4.10. The results are obtained using a 10-fold cross-validation approach.

We can observe from Table 4.10 that both the Word2Vec and fastText embedding approaches reported better prediction accuracy when compared to the GloVe embedding approach. This is mainly due to the fact that both the architectures are neural-network based which able to train the model effectively and extract comparatively highly discriminating sets of features than GloVe architecture.

From the predictive-based embedding models, we observe that the CBOW architecture can extract better sets of features for low similarity datasets when compared to its respective SG architectures. The CBOW architecture, trains the embedding model by predicting the current word from a window of context words, thus minimizing the training error. For high sequence similarity datasets, both CBOW and SG architectures can extract highly discriminating features equally.

Further, it is worth noting that the performance of the fastText-CBOW architecture outperformed other embedding architectures for most of the benchmark datasets. This is mainly because the fastText treats every input word (in the case of structural sequence code) as a set of characters and it is able to train the model effectively. Therefore, we have considered and shortlisted only fastText-CBOW sets of features (1200) for further analysis. The performance variations of all the five embedding architectures across six datasets are shown in Figure 4.3.

4.4.6.3 Performance Analysis of SkipXGram based Feature Sets

Six sets of features were extracted (for X: 0 to 5) on amino acid sequences using the SkipXGram technique. All these feature sets are evaluated individually with the proposed ensemble of classifiers. The overall prediction accuracy obtained using 10-fold cross-validation on all the five datasets are shown in Figure 4.4 where the x-axis represents different values of X (0 to 5), and the y-axis indicates the average prediction accuracy. From Figure 4.4, it can be observed that the three skipped bi-gram feature set (i.e., X=3 or S3G) reported the highest overall prediction accuracy for all the five datasets.

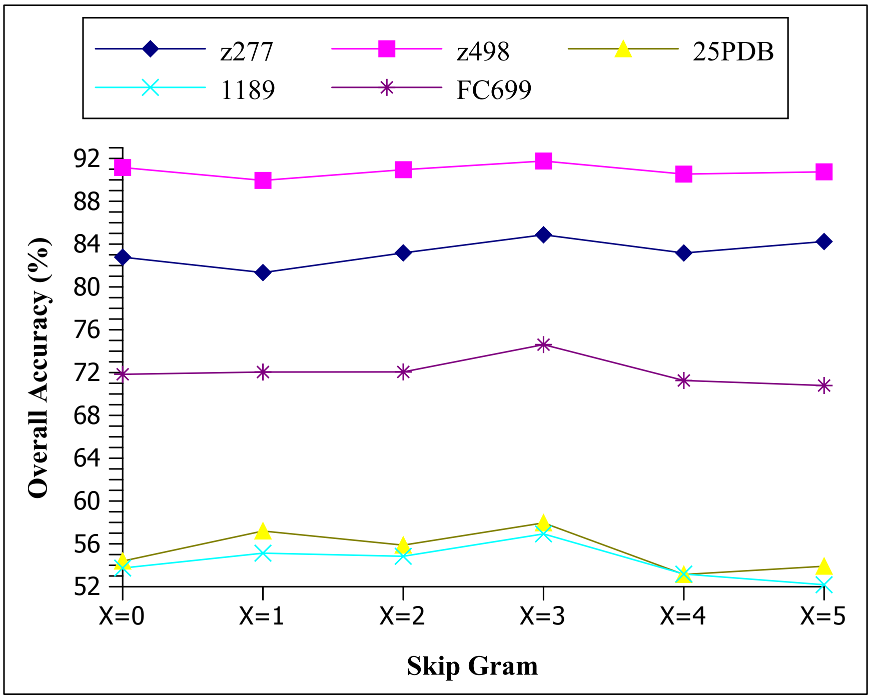


Figure 4.4. Overall Accuracy Variations on Feature Sets extracted from the Amino Acid Sequences for different Skip-gram (i.e., X) values.

A similar analysis was carried out for the other six sets of features that are extracted from secondary structural sequences using the SkipXGram technique. From Figure 4.5,

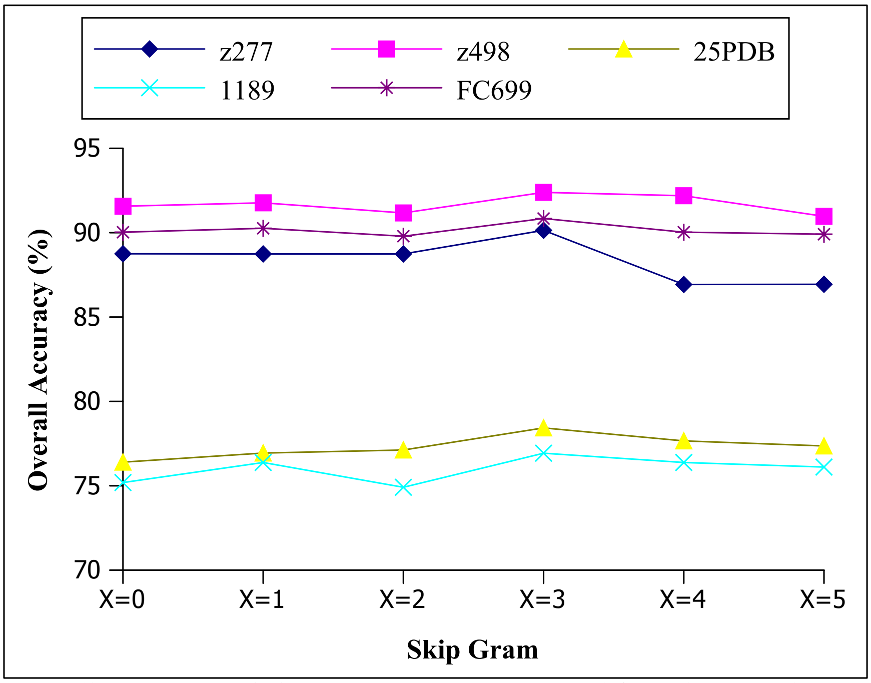


Figure 4.5. Overall Accuracy Variations on Feature Sets extracted from the Secondary Structural Sequences for different Skip-gram (i.e., X) values.

it is observed that the S3G feature set (i.e., X=3) reports the highest overall prediction accuracy when compared to other feature sets. Therefore, we can say that the three skipped bi-gram feature set (i.e. S3G feature set) exhibits highly discriminating features when compared to other skip-gram bi-gram features for the selected five benchmark datasets.

From the Figures 4.4 and 4.5, the following observations are made: (i) the structural class prediction accuracy is high for the high similarity datasets when compared to low similarity datasets. This is mainly due to the fact that high similarity datasets exhibit more discriminating information than low similarity datasets. Further, it is worth noting that the proposed skip-gram bi-gram technique is able to extract discriminating features for the FC699 dataset (which is of low similarity) and thus able to achieve higher prediction accuracy when compared to the other two low similarity datasets. (ii) the feature sets extracted from secondary structural sequences reported higher prediction accuracy across all the benchmark datasets when compared to feature sets from amino acid sequences. Hence, we can say that the secondary structural sequences possess highly discriminating information of structural class when compared to amino acid sequences across all the benchmark datasets.

4.4.6.4 Performance Analysis of the Proposed Feature Extraction Techniques

Along with the nine GS-based feature set, five more feature sets were extracted and shortlisted using the proposed feature extraction techniques. Out of these five sets, three are from fastText-CBOW character embedding (CE) architecture consisting of 1200 features, and the other two sets of features are from the SXG technique (for X=3) consisting of 409 features. We categorized these six feature sets into two groups as E (containing three feature sets) and SXG-GS (containing three feature sets). For both E and SXG-GS, the overall prediction accuracy using 10-fold cross-validation on all the five datasets are recorded and shown in Table 4.11. From Table 4.11 it is observed that the SXG-GS features consistently reported higher prediction accuracy when compared to E features for low sequence similarity datasets. However, E features consistently reported better prediction accuracy than SXG-GS features for high sequence similarity datasets. Since the E technique has the ability to determine similar residues that are in close vicinity in the spatial arrangements of protein sequences, it tends to perform better for high sequence similarity datasets.

Further, the combined set of features (i.e., E-SXG-GS), consisting of 1618 features, reported the highest overall prediction accuracy for all five datasets when compared to individual feature sets. Thus, in this study, the proposed feature modeling combines all six sets (E-SXG-GS) of highly discriminating features.

Table 4.11. The Impact Analysis of the Proposed Feature Extraction Techniques on Benchmark Datasets using Ensemble of Classifiers

Feature Extraction Techniques	**Overall Accuracy (%)**				
	z277	**z498**	**25PDB**	**1189**	**FC699**
SXG-GS	90.20	92.67	78.85	77.19	91.84
CE	91.10	96.55	75.58	76.61	89.05
CE-SXG-GS	**93.55**	**97.58**	**81.82**	**81.12**	**93.93**

4.4.6.5 Performance Analysis of the Proposed Model with State-of-the-art Models

The proposed model effectively combines the proposed feature modeling and an ensemble of three classifiers. The detailed results that are obtained by 10-fold cross-validation on all the datasets are shown in Table 4.12.

The proposed model reported above 93% prediction accuracy for both high similarity datasets, as well as for the FC699 dataset which is a low similarity dataset. For 25PDB and 1189 datasets, the overall prediction accuracy of the proposed model is consistently reported above 81%.

The performance of the proposed model is compared with more than 20 different state-of-the-art models across the five benchmark datasets. It is to be noted that none of the state-of-the-art models have benchmarked their performances on all the five datasets and neither they have made their implementations available to the community. Hence, the performance of the proposed model is compared with these models' results from

Table 4.12. The Various Performance Metrics Result of the Proposed Model on Benchmark Datasets using 10-Fold Cross Validation

Dataset	**Class**	**Sens (%)**	**Spec (%)**	**MCC**	**Overall Accuracy**
z277	All–α	94.29	94.29	0.9225	93.55
	All–β	95.09	95.08	0.9361	
	α/β	96.30	95.12	0.9381	
	$\alpha+\beta$	87.70	89.06	0.8479	
z498	All–α	96.27	98.10	0.9640	97.58
	All–β	98.42	99.20	0.9840	
	α/β	99.26	96.43	0.9700	
	$\alpha+\beta$	96.13	96.88	0.9530	
25PDB	All–α	91.43	88.62	0.8556	81.82
	All–β	82.40	89.24	0.8009	
	α/β	78.62	79.53	0.7291	
	$\alpha+\beta$	74.15	70.32	0.6145	
1189	All–α	88.78	86.08	0.8345	81.12
	All–β	87.41	92.77	0.8576	
	α/β	82.33	81.12	0.7258	
	$\alpha+\beta$	64.70	63.41	0.5336	
FC699	All–α	98.46	97.00	0.9727	93.93
	All–β	94.80	96.59	0.9364	
	α/β	95.76	94.01	0.9069	
	$\alpha+\beta$	75.61	79.49	0.7517	

their respective papers.

The proposed model reported an overall accuracy of 93.55% and 97.58% for high similarity datasets, i.e., z277 and z498 respectively. From Table 4.13 and Table 4.14, it can be observed that the proposed model outperformed all the state-of-the-art models by a maximum margin of around 10% on z277 and around 4% on z498 datasets. The second-best performance for these datasets was reported by Kavianpour et al. Kavianpour and Vasighi (2017) in 2017, where they converted amino acid sequences into binary codes to build cellular automata images. Further, the texture-based features were extracted using the Haralick approach Haralick *et al.* (1973) to predict PSSC. It is worth noting that the activities involved in Kavianpour et al. Kavianpour and Vasighi (2017) to extract features from sequences via automata images is not only computationally expensive but also it is more suitable for high similarity datasets only. The proposed model outperforms the Kavianpour et al. Kavianpour and Vasighi (2017) method for both the datasets. Thus, the proposed feature modeling is efficient and effective in the prediction of the PSSC.

For low similarity datasets, the proposed model reported promising results with an overall accuracy of 81.82%, 81.12% and 93.93% for 25PDB, 1189, and FC699 datasets

Table 4.13. The Performance Comparison (in percentage) of the Proposed Model against State-of-the-art Methods for z277 Dataset

Models	**All–α**	**All–β**	α/β	$\alpha+\beta$	**Overall Accuracy**
IGA-SVM Li *et al.* (2008)	84.30	85.50	92.60	70.70	84.50
IB1 Chen *et al.* (2008)	89.70	88.10	92.20	80.00	87.70
CWT-PCA-SVM Li *et al.* (2009)	85.70	90.20	87.70	80.10	85.90
Information Theoretical Zheng *et al.* (2010)	87.10	80.30	93.80	67.70	83.00
NN-CDM Liu *et al.* (2010*b*)	80.00	86.40	91.60	81.80	85.20
AAC-PSSM-AC Liu *et al.* (2012)	88.60	95.10	97.50	81.50	91.00
LZ-BMKL Mao *et al.* (2013)	92.90	85.30	92.60	69.20	85.60
COMSPA Yu *et al.* (2013)	86.10	87.30	91.30	82.30	87.00
Dehzangi et al. Dehzangi *et al.* (2013*b*)	90.00	93.40	80.00	96.30	90.30
PSSM-LPC Qin *et al.* (2015)	91.40	90.10	92.50	78.40	88.40
Kavianpour et al. Kavianpour and Vasighi (2017)	92.07	93.35	93.47	90.46	92.34
PMCI-RFE Yuan *et al.* (2018)	-	-	-	-	84.43
Proposed Model (this study)	94.29	95.09	96.30	87.77	**93.50**

respectively and it is shown in Table 4.15 - 4.17.

The proposed model outperformed all the state-of-the-art models by a minimum margin of around 3% and the maximum margin of around 22% for the 25PDB dataset as shown in Table 4.15. The best model from the literature, i.e., LCC-PSSM Ding *et al.* (2014) reported an overall accuracy of 79% where 3600 features were extracted using the linear correlation coefficient approach on PSI-BLAST profiles and 278 features were selected to predict the PSSC problem. It is worth noting that the proposed model extracts a 50% lesser number of features than the LCC-PSSM Ding *et al.* (2014) and outperforms the LCC-PSSM Ding *et al.* (2014) model by a factor of around 3%. Moreover, the proposed model's accuracy in predicting the $\alpha+\beta$ class has been improved by more than 10% compared to the LCC-PSSM.

For 1189 dataset, the proposed model outperforms all the state-of-the-art models except the LCC-PSSM Ding *et al.* (2014) as shown in Table 4.16. However, the proposed model's accuracy in predicting the $\alpha+\beta$ class has been improved by more than 6% compared to the LCC-PSSM Ding *et al.* (2014). It is worth noting that many efforts have been carried out to improve the prediction accuracy for $\alpha+\beta$ class prediction and it remains a challenge for the low sequence similarity datasets.

Table 4.14. The Performance Comparison (in percentage) of the Proposed Model against State-of-the-art Methods for z498 Dataset

Models	**All–α**	**All–β**	α/β	$\alpha+\beta$	**Overall Accuracy**
IGA-SVM Li *et al.* (2008)	96.30	93.60	97.80	89.20	94.20
IB1 Chen *et al.* (2008)	94.95	95.83	97.81	94.16	95.74
CWT-PCA-SVM Li *et al.* (2009)	94.40	96.80	97.00	92.30	95.20
NN-CDM Liu *et al.* (2010*b*)	96.30	93.70	95.60	89.90	93.80
Information Theoretical Zheng *et al.* (2010)	95.30	93.70	97.80	88.30	93.80
AAC-PSSM-AC Liu *et al.* (2012)	94.40	96.80	97.80	93.80	95.80
LZ-BMKL Mao *et al.* (2013)	96.30	94.40	96.30	93.80	95.20
COMSPA Yu *et al.* (2013)	95.20	97.60	98.50	90.50	95.40
Dehzangi et al. Dehzangi *et al.* (2013*b*)	95.30	97.60	96.10	97.80	96.80
PSSM-LPC Qin *et al.* (2015)	99.10	96.80	97.80	93.80	96.70
Kavianpour et al. Kavianpour and Vasighi (2017)	96.58	98.49	97.67	96.5	97.31
PMCI-RFE Yuan *et al.* (2018)	-	-	-	-	93.84
Proposed Model (this study)	96.27	98.42	99.26	96.13	**97.58**

Table 4.15. The Performance Comparison (in percentage) of the Proposed Model against State-of-the-art Methods for 25PDB Dataset

Models	**All–α**	**All–β**	α/β	$\alpha+\beta$	**Overall Accuracy**
Specific Tri-Peptides Costantini and Facchiano (2009)	60.60	60.70	67.90	44.30	58.60
AAD-CGR Yang *et al.* (2009)	64.30	65.00	65.00	61.70	64.00
CWT-PCA-SVM Li *et al.* (2009)	76.50	67.30	66.80	45.80	64.00
AADP-PSSM Liu *et al.* (2010*a*)	83.30	78.10	76.30	54.40	72.90
AATP Zhang *et al.* (2012)	81.90	74.70	75.10	55.80	71.70
AAC-PSSM-AC Liu *et al.* (2012)	85.30	81.70	73.70	55.30	74.10
Xia et al. Xia *et al.* (2012)	92.60	72.50	71.70	71.00	77.20
Dehzangi et al. (2013) Dehzangi *et al.* (2013*b*)	86.10	80.80	60.10	80.60	76.70
MEDP Zhang *et al.* (2014)	87.81	78.33	76.01	57.37	74.84
EEDP Zhang *et al.* (2014)	88.04	78.56	78.03	57.14	75.31
LCC-PSSM Ding *et al.* (2014)	91.70	80.80	79.80	64.00	79.00
PSSM-LPC Qin *et al.* (2015)	87.40	81.70	75.10	57.60	75.50
MBMGAC-PSSM Liang *et al.* (2015)	86.70	81.50	79.50	61.70	77.20
Proposed Model (this study)	91.43	82.40	78.62	74.15	**81.82**

For the FC699 dataset, the performance of the proposed model outperforms the state-of-the-art models by a minimum factor of around 2% and a maximum of around 10% as shown in Table 4.17. The best performing model from the literature on the FC699 dataset was reported by Kong et al. Kong *et al.* (2014) in 2014. It can be

Table 4.16. The Performance Comparison (in percentage) of the Proposed Model against State-of-the-art Methods for 1189 Dataset

Models	**All–α**	**All–β**	α/β	$\alpha+\beta$	**Overall Accuracy**
IB1 Chen *et al.* (2008)	65.30	67.73	79.93	40.68	64.65
Specific Tri-Peptides Costantini and Facchiano (2009)	-	-	-	-	59.90
AAD-CGR Yang *et al.* (2009)	62.30	67.70	66.50	63.10	65.20
AADP-PSSM Liu *et al.* (2010*a*)	69.10	83.70	85.60	35.70	70.70
AATP Zhang *et al.* (2012)	72.70	85.40	82.90	42.70	72.60
AAC-PSSM-AC Liu *et al.* (2012)	80.70	86.40	81.40	45.20	74.60
COMSPA Yu *et al.* (2013)	73.25	77.26	76.34	54.91	72.53
Dehzangi et al. Dehzangi *et al.* (2013*b*)	80.20	83.60	44.60	85.40	75.82
EEDP Zhang *et al.* (2014)	84.75	81.97	82.04	47.72	75.00
MEDP Zhang *et al.* (2014)	85.20	84.01	84.43	45.23	75.80
LCC-PSSM Ding *et al.* (2014)	89.20	88.80	85.60	58.50	**81.20**
PSSM-LPC Qin *et al.* (2015)	82.10	86.30	82.60	43.70	74.90
MBMGAC-PSSM Liang *et al.* (2015)	79.80	850	84.70	50.60	76.30
PMCI-RFE Yuan *et al.* (2018)	-	-	-	-	62.37
Proposed Model (this study)	88.79	87.41	82.34	64.73	81.12

Table 4.17. The Performance Comparison (in percentage) of the Proposed Model against State-of-the-art Methods for FC699 Dataset

Models	**All–α**	**All–β**	α/β	$\alpha+\beta$	**Overall Accuracy**
SCPRED Kurgan *et al.* (2008)	-	-	-	-	87.50
Kong et al. Kong *et al.* (2014)	96.20	90.70	96.30	69.50	92.00
PMCI-RFE Yuan *et al.* (2018)	-	-	-	-	82.58
SVM-RFE Yuan *et al.* (2018)	-	-	-	-	83.06
Proposed Model (this study)	98.46	94.80	95.76	75.61	**93.93**

observed that the proposed feature modeling is effective by a factor of 6% in predicting the α+β class when compared to Kong et al. Kong *et al.* (2014).

The accuracy improvement in predicting the α+β class for low similarity datasets is mainly due to the discriminating features which are extracted using SkipXGram and fastText embedding techniques. Thus, we can state that the proposed model is effective in predicting PSSC for both low and high similarity datasets and achieves promising results.

4.4.6.6 Performance Analysis of the Proposed Model on Large-Scale Updated Dataset

In the previous subsection, the performance of the proposed model was evaluated on the benchmark datasets which consisted of fewer volume sequences and did not include newly discovered sequences. To evaluate the robustness of the proposed model, we have carried out experiments on the SCOPe_2.07 dataset. The characteristics of this dataset are available under Latest Low-Similarity High-volume Dataset in section 4.4.1.B.

Table 4.18. The Various Performance Metrics Result of the Proposed Model on Large-Scale Updated Dataset using 10-Fold Cross-Validation

Dataset	**Class**	**Sens (%)**	**Spec (%)**	**MCC**	**Overall Accuracy (%)**
SCOPe_2.07 (Similarity $\leq 25\%$)	All–α	92.44	91.97	0.8941	**81.11**
	All–β	80.07	88.25	0.7892	
	α/β	82.47	79.83	0.7287	
	$\alpha+\beta$	71.48	68.80	0.5810	

The prediction results of the proposed model on SCOPe_2.07 dataset using 10-fold cross-validation is tabulated in Table 4.18. The SCOPe_2.07 dataset consists of a high volume of protein sequences (i.e., 7906) and exhibits a low sequence similarity of $\leq 25\%$. The overall accuracy of the proposed model on the SCOPe_2.07 dataset reported 81.11%, and the results are consistent with the results of the 25PDB dataset which is also of $\leq 25\%$ similarity. By this, we can say that the proposed model performance is consistent and robust even for the large-scale updated dataset.

4.4.6.7 Statistical Significance Analysis

To analyze the statistical significance of the proposed model, we performed paired *t*-test on the overall prediction accuracy among the proposed model and state-of-the-art models. Since no state-of-the-art models were evaluated on all the five benchmark datasets, we have considered six state-of-the-art models (AAC-PSSM-AC Liu *et al.* (2012), Dehzangi et al. Dehzangi *et al.* (2013*b*), PSSM-LPC Qin *et al.* (2015), PMCI-RFE Yuan *et al.* (2018), CWT-PCA-SVM Li *et al.* (2009) and COMSPA Yu *et al.* (2013)) which have reported overall prediction accuracy on a minimum of any three benchmark datasets out of five.

The results of paired *t*-test among the proposed and each of these six state-of-the-art models with a significance level of 5% (i.e. 0.05) are shown in Table 4.19.

From the above-mentioned *t*-test results, the null hypothesis of all the six cases are rejected. Hence, the overall prediction accuracy of the proposed model is statistically significant than that of the state-of-the-art models.

4.5 Summary

The protein secondary structural class prediction plays an important role in analyzing and identifying protein folds, protein tertiary structures, and protein functions. To ad-

Table 4.19. The paired t-test among the Proposed Model and State-of-the-art Models from the Literature

Models	**p-value**	**Null Hypothesis Decision**	**Is the Difference Significant ?**
AAC-PSSM-AC	<0.00001	Reject	Yes
Dehzangi et al.	<0.00001	Reject	Yes
PSSM-LPC	<0.00001	Reject	Yes
PMCI-RFE	<0.00001	Reject	Yes
CWT-PCA-SVM	<0.00001	Reject	Yes
COMSPA	<0.00001	Reject	Yes

dress the PSSC prediction problem, we have proposed a generic approach that predicts the PSSC effectively for both high and low similarity datasets. The proposed model consists of an enhanced feature modeling with the ensemble of three classifiers. The proposed feature modeling consists of three feature extraction techniques such as Character Embedding (CE), SkipXGram (SXGbg), and General Statistical (GS) based feature extraction technique. As a part of feature modeling, various sets of features were extracted using the proposed feature extraction techniques, and finally, six effective sets of features, constituting a total of 1618 features, were shortlisted. The prediction performance of these extracted sets of features was analyzed in detail using an ensemble of three classifiers (i.e., SVM, RF, and GBM). Shortlisted SXGbg features were effective for low-similarity datasets and shortlisted E features were effective for high-similarity datasets. Global GS features enhanced the overall prediction performance. The proposed model reported 93.55% and 97.58% overall accuracy for high similarity datasets namely, z277 and z498 respectively. For low sequence similarity datasets, the proposed model attained 81.82%, 81.12%, and 93.93% on 25PDB, 1189, and FC699 datasets respectively. The performance of the proposed model reported the highest overall accuracy across various benchmark datasets and outperformed all the state-of-the-art models for both low and high similarity datasets. Further, the assessment of the proposed model on the large-scale updated dataset, i.e., SCOPe_2.07 showed that the performance of the proposed model is consistent and robust even for the updated high-volume dataset. From statistical paired t-test results, it has been observed that the overall accuracy of the proposed model significantly outperformed state-of-the-art models. Hence, we conclude that the proposed model is effective and robust in solving the PSSC problem.

In the next chapter, we propose an effective protein fold recognition model. Every structural class of protein is further categorized into various folds based on the tertiary structure of a protein.

Chapter 5

Protein Fold Recognition

In this chapter [1], effective feature extraction and classification model is proposed to address protein fold recognition effectively. Protein folds are the sub-categories under the protein structural class. i.e., every structural class of protein is further categorized into different folds based on the tertiary structure of a protein.

5.1 Datasets

Identification of protein folds for low-similarity datasets is one of the difficult and challenging tasks. Therefore, in this study, we consider only low similarity datasets.

5.1.1 Benchmark Datasets

Three popular and publicly available benchmark datasets namely, DD, EDD, and TG are considered for the performance evaluation of the proposed model.

The first benchmark dataset DD was constructed by the authors' Ding and Dubchak Ding and Dubchak (2001) from SCOP 1.63 version. The DD Ding and Dubchak (2001) dataset consists of 311 protein sequences as a training set that exhibits less than 40% similarity and 383 protein sequences as a test set exhibiting less than 35% similarity. The protein sequences from both sets belong to 27 different folds. The recent studies Paliwal *et al.* (2014*a*); Lyons *et al.* (2016, 2014) on protein fold recognition conducted performance analysis using a 10-fold cross-validation approach on the DD dataset by combining both train and test sets (constituting of total 694 protein sequences).

The second benchmark dataset is an extended version of the DD dataset (EDD) derived from SCOP 1.75 version Andreeva *et al.* (2004). This dataset contains the same 27 folds as the DD dataset with higher volumes of protein sequences constituting a total of 3418 and exhibiting less than 40% similarity.

The authors Taguchi and Gromiha (2007) derived the TG dataset from SCOP 1.73 version. The protein sequences belong to 30 different folds and they exhibit less than 25% similarity. This dataset consists of 1612 protein sequences.

---

[1] The work described in this Chapter has been published in: **Sanjay Bankapur** and Nagamma Patil, "An Enhanced Protein Fold Recognition for Low Similarity Datasets using Convolutional and Skip-Gram Features with Deep Neural Network" in *IEEE Transactions on Nano Bioscience*, vol. 20, no. 1, pp. 42-49, Jan. 2021.

5.1.2 Latest High-volume Low-similarity Datasets

The Structural Classification of Proteins (SCOP) is one of the primary sources of protein sequences with structural annotations. The proposed framework is trained and developed on a high volume of protein sequences that are derived from the latest extended version of SCOPe Fox *et al.* (2013) i.e., SCOPe 2.07[2]. Two main aspects used for sequence extractions from the SCOPe 2.07 database are (i) protein sequences that exhibit less than 25% sequence similarities, and (ii) protein sequences that are part of All-α, All-β, α/β, and $\alpha+\beta$ structural classes only. By this, a total of 7906 protein sequences belonging to 1003 different folds are shortlisted, and 442 out of 1003 folds are having only one protein sequence. In this study, the proposed model is trained and tested using a 10-fold cross-validation approach; therefore, further, we filtered and considered those protein sequences in which every protein fold must exhibit at least ten protein sequences. Using these criteria, the final count of the protein sequences reduced to 6044 with 167 unique folds, and going forward this dataset will be referred to as *25_SCOPe2.07_F167*.

The proposed model is also evaluated and benchmarked on one more derived dataset of SCOPe 2.07. This dataset is constructed by combining all the benchmark protein folds of DD, EDD, and TG datasets and consists of 3262 protein sequences that are belonging to 36 different folds. All the protein sequences of this dataset exhibit less than 25% similarity, and going forward this dataset will be referred to as *25_SCOPe2.07_F36*. The fold names and the number of protein sequences in each fold are listed in Table 5.1.

The overall characteristics of all the datasets that are used in this study are tabulated in Table 5.2.

5.1.3 Data Preparations

As the protein folds are the sub-categories of protein structural class, features of protein sequences and structural sequences are not effective. Moreover, this study only considers low-similarity datasets, evolutionary-based profiles are generated in order to extract distinct relationship-patterns of protein sequences.

Two evolutionary-based profiles such as position-specific scoring matrix (PSSM) and hidden Markov model (HMM) are generated for all the datasets. More details are discussed and available at section 4.2.1

---

[2] `https://scop.berkeley.edu/astral/subsets/ver=2.07`

Table 5.1. The Information of all the Folds and the Frequency of the Protein Sequences in each Fold for the derived 25_SCOPe2.07_F36 Dataset

Number	Class	Fold	Frequency
1	α	Globin-like	26
2		Cytochrome C	26
3		DNA/RNA binding 3-helical bundle	277
4		Four helical up and down bundle	59
5		Four helical cytokines	30
6		EF hand-like fold	43
7		SAM domain-like	58
8		alpha-alpha super helix	136
9	β	Immunoglobulin-like beta-sandwich	297
10		Diphtheria toxin/transcription factors/cytochrome f	43
11		Cupredoxin-like	34
12		Galactose-binding domain-like	47
13		Viral protein domain	5
14		Concanavalin A-like lectins/glucanases	51
15		SH3-like barrel	101
16		OB-fold	137
17		beta-Trefoil	37
18		Trypsin-like serine proteases	19
19		Lipocalins	24
20		Double-stranded beta-helix	86
21		Nucleoplasmin-like/VP	50
22	α/β	TIM beta/alpha-barrel	247
23		NAD(P)-binding Rossmann-fold domains	126
24		FAD/NAD(P)-binding domain	47
25		Flavodoxin-like	106
26		Adenine nucleotide alpha hydrolase-like	49
27		P-loop containing nucleoside triphosphate hydrolases	186
28		Thioredoxin fold	98
29		Ribonuclease H-like motif	124
30		Phosphorylase/hydrolase-like	41
31		S-adenosyl-L-methionine-dependent methyltransferases	88
32		alpha/beta-Hydrolases	75
33		Periplasmic binding protein-like I	42
34	$\alpha+\beta$	beta-Grasp (ubiquitin-like)	93
35		Cystatin-like	79
36		Ferredoxin-like	275

Table 5.2. The Summary of Datasets that are used for Protein Fold Recognition

Dataset	Similarity	Source-Version	Folds	Sequences
DD	$<40\%$	SCOP-1.63	27	694
EDD	$<40\%$	SCOP-1.75	27	3418
TG	$<25\%$	SCOP-1.73	30	1612
25_SCOPe2.07_F167	$<25\%$	SCOPe-2.07	167	6044
25_SCOPe2.07_F36	$<25\%$	SCOPe-2.07	36	3262

5.2 Proposed Methodology:

5.2.1 Feature Extraction:

From the previous chapter outcome, SXGbg features were effective for low-similarity datasets and Embedding features were effective for high-similarity datasets. As this study mainly concentrates on low-similarity datasets, the SXGbg technique is shortlisted and the Embedding technique is ignored.

A global and a local set of features are extracted from evolutionary-based profiles to address protein fold recognition (PFR) effectively. A global set of features are extracted using the proposed convolutional feature extraction technique; whereas, a local set of features are extracted using the proposed SkipXGram bi-gram (SXGbg) technique.

5.2.1.1 Convolutional (Conv) Features

The convolutional feature extraction technique consists of a 2-dimensional (2D) convolutional layer followed by 2D max-pooling layer. For every query sequence, the generated evolutionary-based profile (PSSM or HMM) is of size $L * 20$. L is the length of the query sequence, and 20 indicates the substitution probabilities of 20 amino acid residues. Since query protein sequence length L varies in a dataset, the original profile size is further transformed to a fixed size of 200 * 20 by trimming in case of $L > 200$ or padding with zeros in case of $L < 200$. Going forward the transformed evolutionary profile is referred to as TrP.

In the convolutional layer, the convolutional operation ($\odot$) is performed on TrP by applying 2D-convolutional kernel filter $CKF \ \epsilon \ \mathbb{R}^{ckf_1 \times ckf_2}$ with hyperbolic tangent ($tanh$) as the activation function to obtain 2D convolutional feature map (cfm) as shown in equation 5.1.

$$cfm_{i,j} = tanh\left(CKF \odot TrP_{i:i+ckf_1-1,\, j:j+ckf_2-1} + bias\right) \tag{5.1}$$

From equation 5.1, one cfm is generated by one CKF; Similarly, a set of cfm can be generated by applying n number of CKF in a convolutional layer as shown in equation 5.2

$$cfm\_n = \left\{cfm^1, cfm^2, cfm^3, ..., cfm^n\right\} \tag{5.2}$$

For each cfm^p $(1 \leq p \leq n)$, 2D max-pooling operation mp is performed with the window size $w1 \times w2$ to obtain max-pool feature maps ($mpfm$) as shown in equation

5.3

$$mpfm_{i,j} = mp\left(cfm^{p}_{i:i+w_1-1,\, j:j+w_2-1}\right) \tag{5.3}$$

On applying 2D max-pooling operations to all convolutional feature maps (cfm_n) we obtain n number of max-pool feature maps (denoted as $mpfm\_n$) and it is as shown in equation 5.4

$$mpfm\_n = \left\{mpfm^1,\ mpfm^2,\ mpfm^3,\ \ldots,\ mpfm^n\right\} \tag{5.4}$$

The combination of convolutional and max-pooling operations on evolutionary profiles can identify the most predominant features of a given protein sequence. The generated feature map $mpfm\_n$ is flattened to obtain the $Conv$ feature vector. In this study, the Conv feature vector is of size 112 (more information on Conv feature size is available at section 5.2.3.1).

5.2.1.2 SkipXGram bi-gram (SXGbg) Features

The protein folds are mainly due to the various kinds of interactions among the amino acid residues in close proximity. Bigram technique is a well-known effective approach to extract the local interactions of neighboring amino acids Sharma *et al.* (2013*a*); Lyons *et al.* (2015). The bigram technique can extract features from the conserved regions of the protein sequences. The protein sequences are made up of 20 different amino acids. Hence, the dimension of bigram features is 400 (i.e., 20 × 20).

In the previous section 4.3.2.1, the skip-gram technique successfully explored on linear sequences to extract various sets of local interactions of amino acids, and it was proven to be an effective approach for protein structural class prediction. In this study, we adopted the linear-based skip-gram technique and modified it to extract SXGbg feature sets from 2-dimensional data, i.e., evolutionary-based profiles. To the best of our knowledge, this is the first work to explore the skip-gram technique to recognize the protein folds.

Various levels of amino acid local interactions are captured by skipping a gram (amino acid residue) or a set of consecutive grams. The skip value is denoted by X and six sets of SXGbg features are extracted by varying X values from 0 to 5 (i.e., S0Gbg to S5Gbg) and is represented in equation 5.5.

$$SXGbg(i,\ j) = \sum_{l=1,\ X\epsilon\{0-5\}}^{L-X-1} P_{(l,\ i)} \times P_{(l+X+1,\ j)} \tag{5.5}$$

where, $1 \leq i, j \geq 20$, P is a profile of a dimension $L \times 20$ and L is a length of the query sequence. From six sets, a total of 2400 local interaction features are extracted where each set consists of 400 feature vectors.

5.2.2 Deep Neural Network:

The Conv and SXGbg features that are extracted from the proposed feature extraction techniques are fed into a deep neural network. The proposed deep neural network consists of two fully connected hidden layers, followed by an output layer to predict the protein folds. A total of 2512 feature vectors are fed into the first hidden layer consisting of 512 neurons. The output of the first hidden layer is fed into the second hidden layer, which consists of 128 neurons.

Both hidden layers adopt the hyperbolic tangent function (tanh) as an activation function. The tanh is a smoother zero-centered function whose range lies between -1 to +1, and the equation 5.6 represents the tanh function. The main advantage of tanh function is that it produces zero-centered output, thereby the back-propagation process achieves better training performance for multi-layer neural networks Karlik and Olgac (2011).

$$tanh(x) = \left(\frac{exp^{x} - exp^{-x}}{exp^{x} + exp^{-x}} \right) \tag{5.6}$$

The output layer contains F number of neurons, where F is the total number of unique folds in a given dataset. The PFR being a multi-class classification problem, *softmax* activation function is well suited for PFR. The output of the softmax function Goodfellow *et al.* (2016) is computed by equation 5.7, and the output values are in the range of 0 to 1. The summation of all the softmax output probabilities is equal to 1 and assigns the target fold based on the output with the highest probability.

$$softmax\,(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{F} e^{x_j}} \tag{5.7}$$

In all our experiments, the training error of the proposed deep neural network is calculated using a stochastic gradient descent algorithm Kingma and Ba (2014), and the

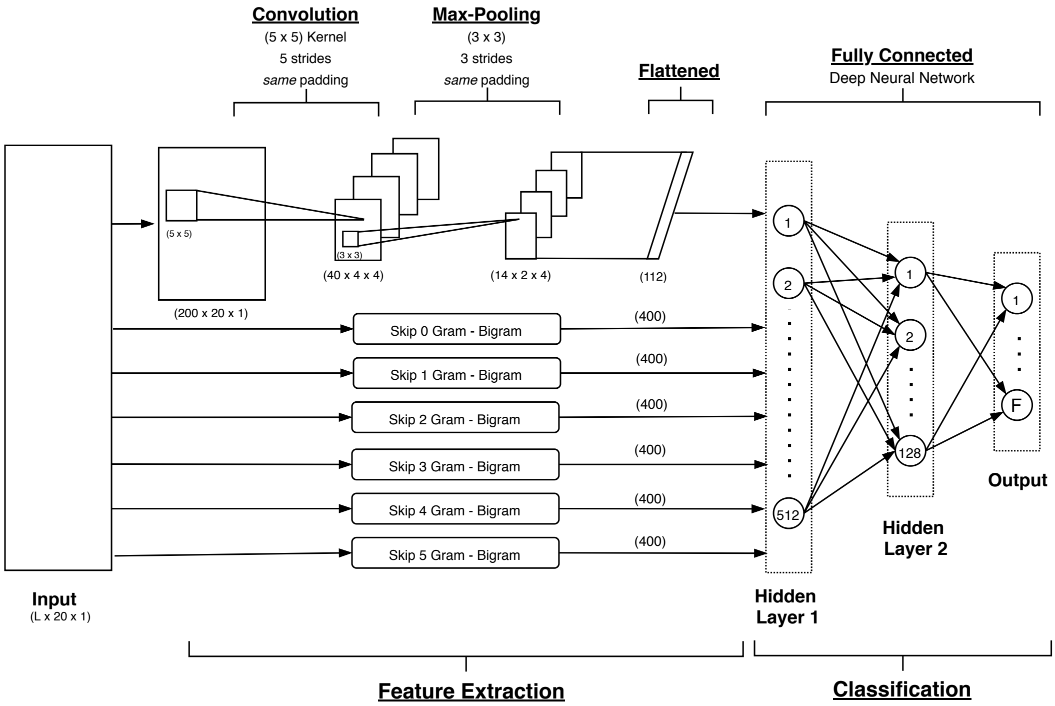


Figure 5.1. The Proposed Framework consisting of Convolutional and Skip-Gram Feature Extraction techniques with Fully Connected Deep Neural Network for PFR.

error is backpropagated to update the neural network weights. The main aim of training the proposed model is to learn the important discriminating patterns by minimizing the cross-entropy loss and the loss is calculated using the equation 5.8, where F is the number of unique folds, *log* is the natural logarithmic function, γ is the L2 regularization hyper-parameter, y_i is the actual fold label for the i^{th} protein sequence, and p_i is the predicted fold label for the same protein sequence. The loss function parameters are optimized using the equation 5.9, where ψ is the parameter rate and β is the learning rate.

$$L(\psi) = -\sum_{i=1}^{F} y_i \left(\log\left(p_i\right)\right) + \gamma \|\psi\|^2 \tag{5.8}$$

$$\psi \leftarrow \psi + \beta \frac{\partial L(\psi)}{\partial \psi} \tag{5.9}$$

5.2.3 The Proposed Model:

Figure 5.1 illustrates the overall framework of the proposed model, which includes two feature extraction techniques namely Convolutional and SkipXGram bi-gram followed by a fully connected deep neural network. In the proposed model, initially, evolutionary profiles such as PSSM and HMM are generated using PSI-BLAST and HHBlits tools, respectively. Then, sets of features are extracted from the profiles using the proposed feature extraction (Convolutional and SkipXGram bi-gram) techniques, as described in the sections 5.2.1.1 and 5.2.1.2. These extracted features are fed into the proposed fully connected deep neural network, as described in section 5.2.2, for protein fold recognition.

5.2.3.1 Parameter Optimization:

The hyper-parameters of the proposed model are optimally tuned for one of the derived datasets, i.e., *25_SCOPe2.07_F167* and the tuned hyper-parameter values are kept constant in the evaluation of other datasets.

The original size of the generated profile is transformed from size $L \times 20$ to 200×20 such that convolutional operation can be performed across protein sequences. In the convolutional layer: the number of convolutional kernel filters (CKF) is varied from 1 to 128, and the optimum results are obtained for four kernels. The size of each filter is varied from 1×1 to 15×15, and the best results are obtained on filter size 5×5. The various values for strides are explored and the optimum results are obtained for five strides with *same* padding. Similarly, in max-pooling layer: the best obtained max-pooling window size is 3×3 with *same* padding and 3 strides. By this, a total of 112 feature vector is extracted from convolutional and max-pooling layers as shown in Figure 5.1. The number of hidden layers is varied from 1 to 10 and the best results are reported for two hidden layers. The number of neurons in hidden layers are explored with various sizes from 4098 to 68, and the best-obtained values are 512 and 128 forming first and second hidden layers, respectively. The best combination of the parameter values as mentioned above is identified using grid-search, and these parameter values are kept constant across all the experiments.

5.3 Results and Discussions

In this section, first, we will highlight the experimental setup followed by the performance analysis of the proposed model. The performance analysis is carried out in two stages: Initially, an ablation study of various sets of features extracted on the two derived datasets. Later, the proposed framework is evaluated on three benchmark datasets and compared against the best models from the literature.

5.3.1 Experimental Setup

All the experiments were carried out on an Ubuntu-based server having 128 GB RAM, 56 cores of Intel Xeon processors, 3TB hard drive, and two NVIDIA Tesla M40 GPUs. The proposed SXG-bg feature extraction technique is implemented in Python 3, and the proposed Convolutional feature extraction technique with fully connected layers is implemented using Keras (provided by Tensorflow) Abadi *et al.* (2016). The performance of the proposed model is evaluated on all the datasets using an accuracy metric. Accuracy is the ratio of correctly recognized folds of protein sequences (both true positives and true negatives) to the total number of protein sequences, and it is as shown in the equation 5.10.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.10}$$

Where, TP, TN, FP, and FN are the total number of true positives, true negatives, false positives, and false negatives, respectively. In this study, all the experiments are carried out using a 10-fold cross-validation approach.

5.3.2 An ablation study of feature sets on the derived datasets of SCOPe 2.07

In this section, two-levels of impact analysis are carried out on the feature sets that are extracted from the derived datasets. First, an analysis on evolutionary-based profiles (PSSM vs HMM) is presented and followed by the importance of global and local feature sets. Features are extracted from two evolutionary-based profiles such as PSSM and HMM, from the two derived datasets, i.e., 25_SCOPe2.07_F167 and 25_SCOPe2.07_F36. The detailed information of these datasets is available in section 5.1.2. A total of seven sets of features are extracted in which one is a global feature set (Conv) extracted using the proposed Convolutional approach and the rest six sets are local feature sets (i.e., S0Gbg to S5Gbg) extracted using the proposed SXGbg technique. The extracted feature set is fed into the proposed deep neural network to classify the query sequence to its respective folds and the obtained PFR accuracy on the extracted feature sets is tabulated in Table 5.3.

From the Table 5.3, the first seven rows represent the respective individual PFR accuracy of seven extracted feature sets followed by the combined set of all the local features (i.e., S0Gbg to S5Gbg, and it is referred to as SXGbg feature set consisting of 2400 feature vector) and the last row represents the combination of all the global and local feature sets (referred as Conv+SXGbg features set).

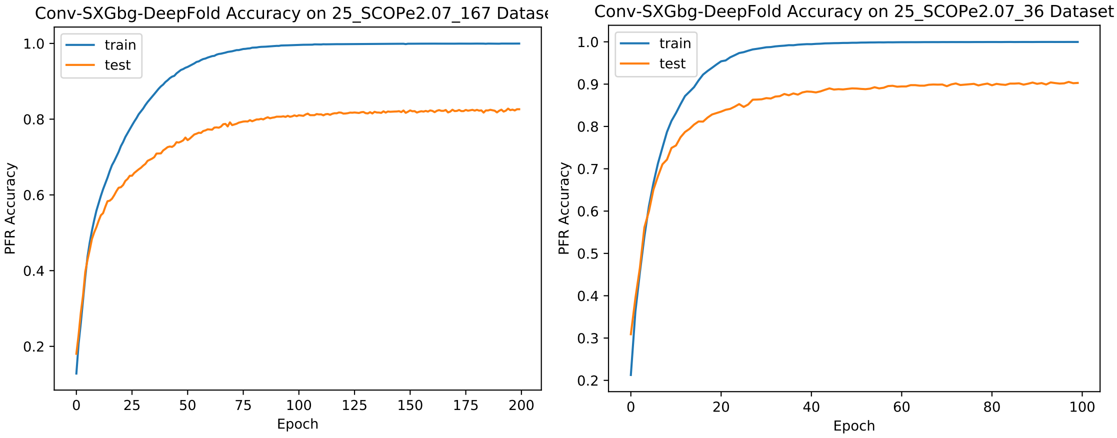
PSSM vs HMM: From the Table 5.3, it can be observed that HMM-based features boosted more than 20% of PFR accuracy when compared to PSSM-based features for the 25_SCOPe2.07_167 dataset. Similarly, for the 25_SCOPe2.07_36 dataset, HMM-based features recorded 15% more PFR accuracy when compared to PSSM-based features. The improvement is mainly due to the fact that HHBlits alignment tool Remmert *et al.* (2012) generates HMM profiles effectively by identifying and aligning remote homology blocks accurately for low similarity datasets when compared to PSI-BLAST Altschul *et al.* (1997). Moreover, HHBlits being a profile-profile alignment tool, successfully captures rich evolutional information of low similarity query sequences that helps to recognize its fold accurately.

Global and Local Features: From the Table 5.3, it can be observed that the individual local feature set (S0Gbg or S1Gbg or ... or S5Gbg) is less effective in PFR when compared to the combination of all the local feature sets (SXGbg). There is a minimum of 9.4% and 6.8% absolute improvement over individual local sets for PSSM and HMM-based SXGbg feature sets respectively on the 25_SCOPe2.07_167 dataset. Similarly, for the 25_SCOPe2.07_36 dataset, the PFR accuracy of all the combined local feature sets (SXGbg) improved by at least (absolute) 11.2% and 4.5% on PSSM and HMM-based features respectively. The PFR accuracy improvement in the SXGbg feature set is mainly because a various local amino acid interaction is effectively captured in each local set and every local feature set is complimentary to rest of the local feature sets. Further, it is worth observing that the combination of global and local feature sets (Conv+SXG) are even more effective in predicting the protein folds and recorded a minimum improvement of (absolute) 2.4% and 1.2% on 25_SCOPe2.07_167 and 25_SCOPe2.07_36 datasets respectively when compared to either global or local feature set. Both global and local feature sets are compliment to each other in the enhancement of PFR performance.

Table 5.3. The Performance (in percentage) of Various Feature Sets On Derived Datasets

Features	**Features Size**	**25_SCOPe2.07_167**		**25_SCOPe2.07_36**	
		PSSM	**HMM**	**PSSM**	**HMM**
Conv	112	21.9	42.1	35.5	58.7
S0Gbg	400	41.2	74.1	59.2	85.2
S1Gbg	400	41.0	74.1	59.9	85.1
S2Gbg	400	40.0	73.2	58.7	83.3
S3Gbg	400	39.6	73.4	55.6	85.7
S4Gbg	400	37.7	71.6	54.5	83.1
S5Gbg	400	36.9	70.7	53.5	84.3
SXGbg	2400	50.6	80.9	71.1	90.2
Conv+SXGbg	2512	53.0	**83.3**	76.3	**91.4**

The HMM-based Conv and SXGbg features reported the best performance, i.e., 83.3% for the 25_SCOPe2.07_F167 dataset and 91.4% on the 25_SCOPe2.07_F36 dataset. Thus, all the further experiments are carried out with the combination of Conv and SXGbg feature sets that are extracted only from HMM profiles, and fold recognition



(a) On 25_SCOPe207_F167 dataset

(b) On 25_SCOPe207_F36 dataset

Figure 5.2. The correlation of train and test accuracy of the proposed model for different epochs

is performed using the proposed deep neural network. Henceforth the proposed framework will be referred to as Conv-SXGbg-DeepFold.

All the experiments of the proposed Conv-SXGbg-DeepFold model are trained for 200 epochs with early stopping condition. The correlation of the train and test accuracy of the proposed Conv-SXGbg-DeepFold model for different epochs on 25_SCOPe2.07_F167 and 25_SCOPe2.07_F36 datasets are shown in Figure 5.2. From 5.2a and 5.2b, it is observed that the training accuracy increases as the epochs increase and the training accuracy attains a maximum value 1, indicating that the model is trained nearly to 100% accuracy exhibiting low bias. Further, it is also observed that as the training accuracy increases, the testing accuracy also increases, implying the convergence of the proposed model. Both the training and testing accuracies are stabilized over 200 and 100 epochs for 25_SCOPe2.07_F167 and 25_SCOPe2.07_F36 datasets respectively.

5.3.3 Comparison with state-of-the-art models:

To demonstrate the effectiveness of the proposed Conv-SXGbg-DeepFold, the fold recognition results on three low similarity benchmark datasets are compared with the state-of-the-art models. PFR results of the state-of-the-art models and the proposed Conv-SXGbg-DeepFold model are tabulated in Table 5.4. The experiments are carried out using 10-fold cross-validations. The state-of-the-art models' results on all benchmark datasets are taken from their published work and (-) indicates the unavailability of the result.

The proposed model reported PFR accuracy of 85.9%, 95.8%, and 88.8% on DD, EDD, and TG datasets, respectively. The proposed Conv-SXGbg-DeepFold is the first model to achieve PFR accuracies over 85% on DD, 95% on EDD, and 88% on TG datasets.

Table 5.4. The Performance Comparison (in percentage) of the Proposed Conv-SXGbg-DeepFold Model against the State-of-the-art Models on DD, EDD and TG benchmark datasets.

Models	**References**	**DD**	**EDD**	**TG**
AAC	Ding and Dubchak (2001)	45.1	40.9	32.0
AAC+HXPZV	Ding and Dubchak (2001)	47.2	40.9	36.3
Taguchi & Gromiha	Taguchi and Gromiha (2007)	51.0	46.9	36.2
PF1	Ghanty and Pal (2009)	50.6	50.8	38.8
PF2	Ghanty and Pal (2009)	48.2	49.9	38.8
PF	Ghanty and Pal (2009)	53.4	55.6	43.1
ACCFold	Dong *et al.* (2009)	70.1	87.6	-
TAXFOLD	Yang and Chen (2011)	71.5	86.9	-
CONS-AAC	Sharma *et al.* (2013*a*)	59.2	61.9	44.0
Mono-gram	Sharma *et al.* (2013*a*)	69.6	76.9	58.8
Bi-gram	Sharma *et al.* (2013*a*)	74.1	84.5	68.1
Alignment method	Lyons *et al.* (2014)	74.7	90.2	74.0
k-AAP	Paliwal *et al.* (2014*a*)	76.1	90.6	77.0
Paliwal et al.	Paliwal *et al.* (2014*b*)	-	86.2	72.5
PSSM-SPINE-S	Dehzangi *et al.* (2014)		88.2	73.8
Saini et al.	Saini *et al.* (2015)	76.7	89.9	74.5
HMM-Bigram	Lyons *et al.* (2015)	79.4	92.6	83.1
HMM-Trigram	Lyons *et al.* (2015)	81.8	93.8	86.0
PHMM-DP	Lyons *et al.* (2016)	82.7	92.9	85.6
MF-SRC	Yan *et al.* (2017)	78.6	86.2	79.8
OVAOVO-DKELM	Ibrahim and Abadeh (2018)	62.7	-	75.8
Conv-SXGbg-DeepFold	This Study	**85.9**	**95.8**	**88.8**

From Table 5.4, it can be observed that the proposed model outperforms all the state-of-the-art models in protein fold recognition across all three benchmark datasets. The proposed model's fold recognition results are improved by a minimum of 5% on DD, 2% on EDD, and 3% on TG datasets when compared to the next-best model, i.e., HMM-Trigram Lyons *et al.* (2015). It is worth mentioning that the HMM-Trigram model Lyons *et al.* (2015) utilizes 8000 features; whereas, the proposed model utilized relatively 68% fewer features (i.e., a total of 2512 features only) when compared to HMM-Trigram Lyons *et al.* (2015) model.

Discussion: The trigram feature extraction technique is one of the effective approach to solve the protein fold recognition Paliwal *et al.* (2014*b*); Lyons *et al.* (2015). Even though the size of the trigram features (i.e., 8000) is large when compared to SXGbg features (i.e., 2400), the trigram technique falls short in extracting discriminating features. This is because by nature trigram features exhibit redundant information due to higher overlapping of amino acid interactions. It is well known that as the number of features increases the training and testing time of the model also increases. Whereas, the proposed feature extraction approach can extract and capture various levels of local amino acid interactions as well as global amino acid interactions effectively. Hence, we can say that the proposed feature extraction techniques are effective in extracting highly discriminating information and efficient in training the model.

5.3.4 Statistical Significance Analysis:

The proposed Conv-SXGbg-DeepFold model outperformed all the state-of-the-art models by a minimum margin of 4%, 2%, and 3% on DD, EDD, and TG datasets, respectively. To demonstrate the significance in the performance improvement of the proposed model, a statistical paired t-test is carried out on the protein fold recognition accuracies among the proposed Conv-SXGbg-DeepFold model with the two next-best models from literature such as PHMM-DP Lyons *et al.* (2016) and HMM-Trigram Lyons *et al.* (2015).

A null hypothesis H_0 on a significance level of 5% (i.e., 0.05) is defined as there is no significant difference among the performances of the proposed Conv-SXGbg-DeepFold and the two next-best models from the literature. The H_0 is rejected when $p < 0.05$ indicating there is a statistically significant difference in the results. Otherwise, the H_0 is retained as there is no significant difference in the results. The statistical paired t-test results are shown in Table 5.5.

Table 5.5. The paired t-test among the Conv-SXGbg-DeepFold and two next-best models from the literature

Model	**Dataset**	**p-value**	H_0 **Decision**	**Is the Difference Significant?**
PHMM-DP Lyons *et al.* (2016)	DD	0.00230	Reject	Yes
	EDD	0.00001	Reject	Yes
	TG	0.00145	Reject	Yes
HMM-Trigram Lyons *et al.* (2015)	DD	0.00043	Reject	Yes
	EDD	0.00010	Reject	Yes
	TG	0.00338	Reject	Yes

From Table 5.5, we can see that the proposed Conv-SXGbg-DeepFold model rejected the null hypothesis on both state-of-the-art models across all three datasets. Hence, we claim that the proposed Conv-SXGbg-DeepFold model is effective in solving the protein fold recognition across various low similarity datasets.

5.4 Summary

Protein fold recognition is one of the important steps in discovering the protein tertiary structure and its functions. The protein fold recognition of low similarity sequences is still considered to be a challenging task in computational biology. Numerous models have been published over a decade to solve this problem effectively. However, most of the models reported the fold recognition accuracy below 80% on benchmark datasets and a limited number of models are above 80% accuracy. In this study, a combination of Convolutional (Conv) features and SkipXGram bi-gram (SXGbg) features have been extracted from the proposed feature extraction techniques, and fold recognition has been performed using the proposed deep neural network. The performance of the proposed Conv-SXGbg-DeepFold model has been benchmarked on two derived datasets from the latest extended version of SCOPe_2.07 such that the derived datasets contain a high volume of protein sequences with a low similarity of less than 25% and belonging to more than 35 different folds. The proposed model reported 91.4% fold accuracy on one of the derived datasets that belong to 36 different benchmark folds. The performance of the proposed model has been evaluated against the state-of-the-art models on three benchmark datasets, and the results of the proposed model outperformed all the state-of-the-art models. The proposed model reported 85.9%, 95.8%, and 88.8% on DD, EDD, and TG benchmark datasets respectively. The performance of the proposed model improved by 5% to 23%, 2% to 19%, and 3% to 30% on DD, EDD, and TG datasets, respectively when compared to the best models from the literature. A statistical significance test was performed on the improved results by conducting paired t-test with a significance level of 5%, and the results of the statistical test showed that the improvement of the protein fold recognition accuracies of the proposed model was significant. From all the conducted experiments, we conclude that the proposed Conv-SXGbg-DeepFold model is effective in solving the protein fold recognition problem.

In the next chapter, effective multi-label protein sub-chloroplast localization prediction models are discussed.

Chapter 6

Protein Subcellular Localization Prediction

A chloroplast is one of the most classic organelles in algae and plant cells. Identifying the locations of chloroplast proteins in the chloroplast organelle is an important as well as a challenging task in deciphering their functions. Protein Sub-Chloroplast Localization (PSCL) prediction is a level-more microscopic problem of subcellular localization and it is considered as a multi-label problem. In this chapter, two novel models are proposed to solve a multi-label PSCL prediction problem. First model [1], utilizes Binary Relevance (BR) approach to solve multi-label PSCL prediction and the second model [2], solves multi-label PSCL using a deep learning framework.

6.1 Datasets

In this study, two publicly available datasets, such as Benchmark and Novel have been considered to evaluate the proposed model.

The Benchmark dataset was derived from the May-2013 release of the UniProtKB/Swiss-Prot database by the authors Wang *et al.* (2015). It contains a total of 578 protein sequences exhibiting $< 40\%$ similarity among the sequences. These sequences are distributed among five sub-locations of chloroplast organelle such as Envelope, Lumen (Thylakoid-lumen), Membrane (Thylakoid-membrane), Plastoglobule, and Stroma. Out of 578 sequences, 556 sequences belong to one sub-chloroplast location, 21 sequences belong to two sub-chloroplast locations, and one sequence belongs to three sub-chloroplast locations. Thus, Benchmark is a multi-label dataset, i.e., there are 22 sequences in which each sequence belongs to more than one sub-chloroplast location.

The protein sequences that were added to the Swiss-Prot database from June-2013 to Nov-2015 are considered as the source to derive the Novel dataset by Wan et al. Wan *et al.* (2016*a*). The novel dataset consists of 122 protein sequences that were distributed among four sub-locations of chloroplast organelle such as, Envelope, Lumen (Thylakoid-lumen), Membrane (Thylakoid-membrane), and Stroma. Out of 122 sequences, 113 and nine sequences belong to one and two sub-chloroplast locations respectively. Thus, Novel is also a multi-label dataset.

---

[1] The work described in this Chapter has been submitted for possible publication as Abhilash Venkatesh, Shrinivas V. Shanbhag, **Sanjay Bankapur**, and Nagamma Patil, ”Multi-Label Protein Sub-Chloroplast Localisation Prediction using Binary Relevance Framework and Machine Learning Techniques”. *International Journal of Data Mining and Bioinformatics, Inderscience*. (**Communicated**)

[2] The work described in this Chapter has been published in: **Sanjay Bankapur** and Nagamma Patil, ”An Effective Multi-Label Protein Sub-Chloroplast Localization Prediction by Skipped-grams of Evolutionary Profiles using Deep Neural Network” in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, (In Press, 2020).

The data characteristics and sequence frequencies for each sub-chloroplast locations are shown in Table 6.1.

Table 6.1. Data Characteristics of Multi-Label Sub-Chloroplast Datasets

Dataset	**Locations (Labels)**	**Sequence Frequencies**			
		Overall	**One Label**	**Two Label**	**Three Label**
Benchmark Wang *et al.* (2015)	5	578	556	21	1
Novel Wan *et al.* (2016*a*)	4	122	113	9	0

6.2 Data Preparations

In this study, from a query protein sequence, protein secondary structural sequence is generated using the PSIPRED tool and two evolutionary-based profiles such as PSSM and HMM are generated. More details about PSIPRED, PSSM, and HMM are discussed and available at section 4.2.1

For both the multi-label datasets, secondary structural sequences, and two evolutionary-based profiles were generated.

6.3 Protein Sub-Chloroplast Localization Prediction using Binary Relevance

In this section, we propose a novel multi-label prediction model using an effective feature modeling with BR framework to classify multi-label sub-chloroplast proteins to its respective localization.

6.3.1 Feature Modeling

Mainly two types of feature extraction studies have been carried out, i.e., the evolutionary profile-based features which include PSSM and HMM, and the other being embedding-based features via Word2Vec on both amino acid and secondary structural sequences.

6.3.1.1 Evolutionary Profile-based Features

As mentioned in section 6.2, two evolutionary profiles are generated -i) PSSM of size $L \times 42$ containing two sub-matrices of size $L \times 20$ i.e., PSSM-lo and PSSM-ls ii) HMM profile of size $L \times 30$ of which sub-matrix $L \times 20$ are used for further study, where L is the length of the protein sequence. Following n-gram extraction techniques are performed on these evolutionary profile matrices (EPM) to extract features :

1. Mono-gram: Mono-gram features Taguchi and Gromiha (2007) are extracted from 20 column profile alignment matrix by normalizing the column wise sum of

the probability values and it is as shown in the below equation.

$$Mono\text{-}gram(AA_j) = \sum_{i=0}^{L-1} EPM_{i,j} \tag{6.1}$$

where i denotes the i^{th} amino acid in the sequence, L denotes the length of amino acid sequence, j denotes the amino acid column in evolutionary profile matrix (EPM) of amino acid sequence and AAj denotes the mono-gram feature. As $0 \leq j < 20$, the matrix yields vector feature size of 20.

2. Bi-gram: Bi-gram features have been found to be successful in protein-related predictions Sharma *et al.* (2013*a*); Zaman *et al.* (2017). Bi-gram features are extracted from normalized evolutionary-based profiles and are generated using the following equation:

$$Bi\text{-}gram(AA_j, AA_k) = sum_{i=0}^{L-1} EPM_{i,j}\ EPM_{i+1,k} \tag{6.2}$$

where j and k denote the amino acid column pairs for which the bi-gram is calculated and $(AA_j$,$AA_k)$ denotes the bigram feature. As, $0 \leq j, k < 20$, the number of $(AA_j$,$AA_k)$ features generated from each matrix is 400.

6.3.1.2 Character Embedding Features

CE-based features are extracted from protein sequences and secondary structural sequences. More details were discussed in section 4.3.2.2. Using this technique, two sets of feature vectors (each of size 400) are extracted from protein sequence and structural sequence respectively. Going forward we denote CE-based features as Word2Vec features of size 800.

6.3.2 Feature Selection

From the above feature extraction approaches, a well-blended feature set of 1200 features (400 PSSM-lo bi-gram + 800 Word2Vec) are shortlisted (as mentioned in section 6.3.1). To find the appropriate features which are high in discriminating sub-chloroplast locations, we performed feature selection on 1200 features. Feature selection is performed label-wise, i.e., the best features for each label are selected. As a result, the features might differ between each label. Genetic Algorithm (GA) is one of the robust methods for feature selection in subcellular localization study Wang *et al.* (2015); Lin *et al.* (2013).

Before performing label specific feature selection, data transformation of the multi-label dataset is carried out. Data-feature dataset D is replicated into data-feature set D_i

for $i = 1$ to k, (where k is the number of labels) such that feature set of each protein sequence S_j having the label L_i is labelled as 1 else labelled as 0 (Label i vs. rest) as shown in Figure 6.1. GA is applied to each D_i to optimize for label L_i with maximizing the accuracy of a binary classifier as the fitness function. After application of GA, each data-feature set D_i yields label specific feature dataset $D_i^{'}$ containing a feature subset of D and not necessarily $fe(D_i^{'}) = fe(D_j^{'})$ holds where $fe(X)$ denotes features of dataset X, $1 \leq i, j \leq k$ and $i \neq j$.

6.3.3 Multi-label Classification

In this work, the Binary Relevance (BR) approach is adopted to address the multi-label protein sub-chloroplast localization (PSCL) prediction problem. Finding an optimal k value in the adaptive approach, such as ML-KNN is a challenging task. Moreover, as the number of multiple location proteins is limited, the label power set approach is not a suitable option.

6.3.3.1 Classifiers

To find the optimum base classifier for the BR framework, we have considered seven state-of-the-art classifiers such as Naive Bayes (NB), Logistic Regression (LR), Gradient Boosting Machine (GBM), Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), K-Nearest Neighbour (k-NN), and Random Forest (RF) are considered. The details of the NB classifier are provided below and the details of the other six classifiers were discussed in section 4.3.3.

Naive Bayes (NB): The Bayesian probabilistic classifier is based on the assumption that each feature makes an independent and equal contribution to the outcome. Naive Bayes finds the posterior probability of each class given a data point by calculating the prior probability of the class and the likelihood of data points belonging to that class. Gaussian naive Bayes classification is a case of naive Bayes method with an assumption of having a Gaussian distribution on attribute values given the class label.

From these classifiers, a quantitative analysis is carried out to shortlist the best performing base classifier for the BR framework.

6.3.4 Proposed model

From literature, models such as LIFT Zhang and Wu (2014), MultiP-SChlo Wang *et al.* (2015), and EnTrans-Chlo Wan *et al.* (2016*b*) showed that the choice of feature extraction, feature selection/reduction approaches and classification technique for multi-label will significantly affect the performance of multi-label protein localization prediction.

Therefore, in this study we proposed two variations of BR model. First, with GA-based feature selection on BR framework consisting SVM as base classifier and will be denoted as GA-BiSVM. The other is without any feature selection and will be denoted as BiSVM.

BiSVM: BiSVM model is Binary Relevance (BR) with SVM as base classifier. From section 6.3.5.4, it is found that the performance of SVM is better among the various state-of-the-art classifiers. Therefore, SVM is chosen as the base classifier of BR. In BR, multiple single-label data transformation techniques is used to deal with multi-label classification. Therefore, each protein sequence (S_j) in data-feature set D is replicated as shown in the steps 2-11 of Algorithm 6.1 to obtain k number of transformed data-

Algorithm 6.1. : Proposed BiSVM algorithm to predict multi-label sub-chloroplast location

```
Input: Protein sequence ($S$) dataset as input
Output: Protein sequences with one or more sub-chloroplast location information as output.
1:  PSSM profiles are generated and bi-gram features are extracted for each protein sequence. The extracted features of all protein sequences $S$ are collectively named as data-feature set $D$.
2:  For multi-label prediction, the generated data-feature set $D$ is transformed as follows :
3:  for each protein sequence $S_j$ in data-feature set $D$ do
4:      for each label $L_i$ in label-set $L_1, L_2, ..., L_k$ (where k is the number of unique labels) do
5:          if $S_j$ has label $L_i$ then
6:              Replicate the feature set of $S_j$ with label as 1 in data-feature set $D_i$.
7:          else
8:              Replicate the feature set of $S_j$ with label as 0 in data-feature set $D_i$.
9:          end if
10:     end for
11: end for
    Training phase:
12: for each label $L_i$ in label-set $L_1, L_2, ..., L_k$ do
13:     Train the binary SVM classifier $C_i$ on dataset $D_i$
14: end for
    Testing phase:
15: for each protein sequence $S_j$ from test set do
16:     Features set $Sj$ is fed into all binary SVM classifiers $C_i$ to obtain output $a_i$ , where $a_i$ is 0 or 1.
17:     The outputs of all classifiers $C_i$ are combined to obtain a vector $V_j = [a_1, a_2, ...., a_k]$.
18: end for
```

feature sets D_i. BR consists of k (no. of labels) binary classifiers where each classifier C_i (where $1 \leq i \leq k$) classifies each protein sequence S in data-feature sets D_i independently to whether or not (1 or 0) it belongs to label L_i. The outcome of all the classifiers C_i are combined to obtain the multi-label prediction output in the form of vector V_i, where $V_i = [a_1, a_2, ..., a_k]$, a_j is binary classification (a_j is 1 or 0) of classifier C_i. The classification and the output generation steps are highlighted in the Training and Testing phase of Algorithm 6.1.

GA-BiSVM: Label specific features using Genetic Algorithm (GA) successfully enhanced the prediction accuracy of multi-label subcellular localization problem Lin *et al.* (2013); Wang *et al.* (2015). In this regard, we adopted GA to select and analyse the label specific feature sets. GA-BiSVM is a two-stage classification model, and it is, as shown in Figure 6.1. At the first stage, GA based feature selection is performed on

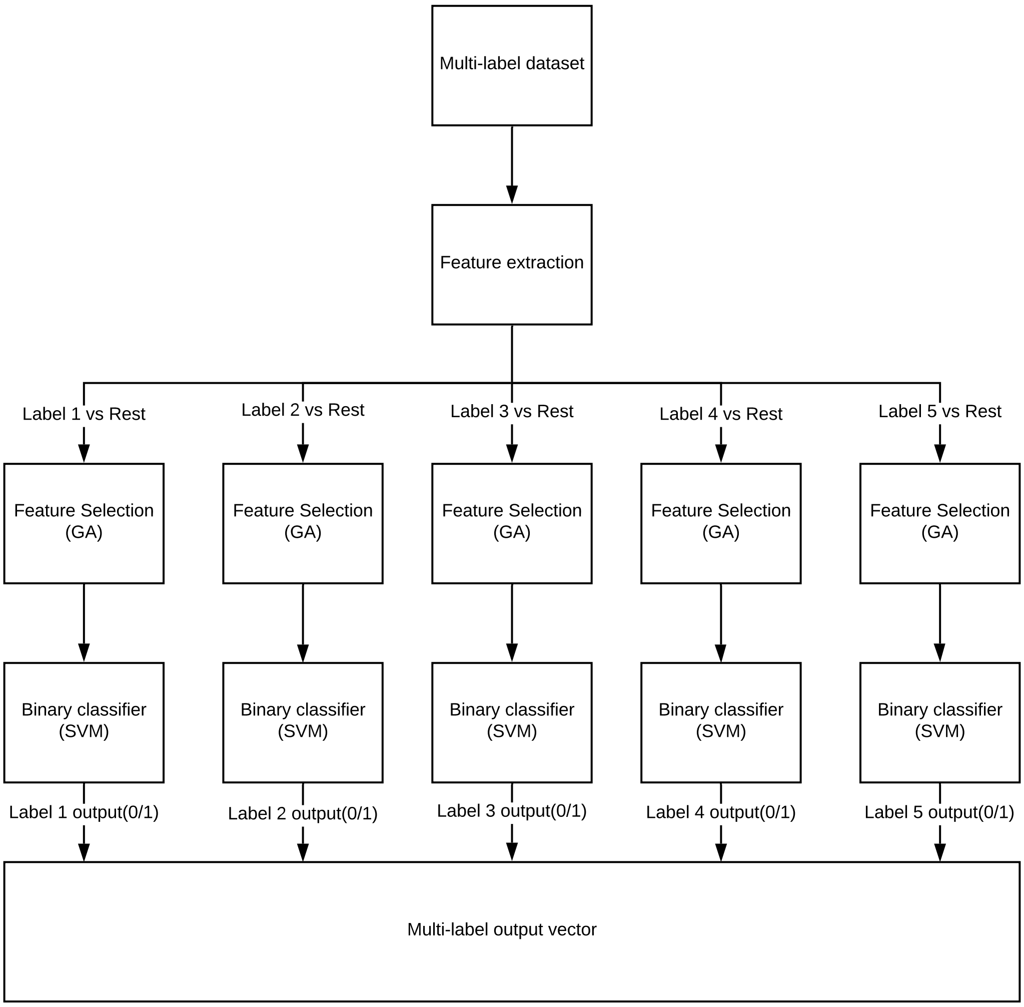


Figure 6.1. Two-stage GA-BiSVM Classification Model for PSCL Prediction

all the transformed data-feature sets D_i to obtain label specific feature datasets $D_i^{'}$, as mentioned in section 2.3. In the second stage, a BR is adopted to perform multi-label classification using label specific feature dataset $D_i^{'}$. The detailed architecture of the proposed approach is shown in Figure 6.1.

6.3.5 Results and Discussion

6.3.5.1 Experiment Setup

The Mono-gram, Bi-gram, and Word2Vec feature extraction techniques are implemented in Python 3 and the proposed multi-label BR framework is also implemented in Python 3. All the experiments of the proposed model are carried out on an Ubuntu-based server having 128 GB RAM, 56 cores of Intel Xeon processors, two NVIDIA Tesla M40 GPUs, and a 3TB hard drive.

6.3.5.2 Evaluation Metrics

To measure the performance of multi-label PSCL classification requires more sophisticated metrics than single-label classification. In this study, six popular metrics such as Overall Actual Accuracy (OAA), Accuracy (Acc), Precision, Recall, F1-score (F1), and Grand Mean are considered and they are defined as follows:

Let $AL(P_i)$ and $PL(P_i)$ be an actual label set and the predicted label set for the i^{th} protein sequence from a given dataset respectively. These metrics are defined as follows:

$$OAA = \frac{1}{N}\sum_{i=1}^{N}\Delta[AL(P_i), PL(P_i)] \tag{6.3}$$

where,

$$\Delta[AL(P_i), PL(P_i)] = \begin{cases} 1, & if AL(P_i) = PL(P_i) \\ 0, & otherwise \end{cases}$$

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\left|AL(P_i) \cap PL(P_i)\right|}{\left|AL(P_i) \cup PL(P_i)\right|}\right) \tag{6.4}$$

$$Precision = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\left|AL(P_i) \cap PL(P_i)\right|}{\left|PL(P_i)\right|}\right) \tag{6.5}$$

$$Recall = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\left|AL(P_i)\cap PL(P_i)\right|}{\left|AL(P_i)\right|}\right) \tag{6.6}$$

$$F1 = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{2\left|AL(P_i)\cap PL(P_i)\right|}{\left|AL(P_i)\right| + \left|PL(P_i)\right|}\right) \tag{6.7}$$

$$GrandMean = \frac{(OAA + Acc + Precision + Recall + F1)}{5} \tag{6.8}$$

where, $|.|$ indicates the number of elements count in a set, intersection represents the intersection of sets, and union represents the union of sets. For all metrics, the higher the values, the better the prediction performance.

6.3.5.3 Cross Validation

In order to perform a fair evaluation comparison of the performance of the proposed model against state-of-the-art models, we followed the same cross-validation approach as the state-of-the-art models in this study.

All the experiments on the Benchmark datasets are carried out using a leave-one-out cross-validation approach. The generalizability of the proposed model is further verified using an independent test set approach, in which the proposed model is trained on the Benchmark dataset and tested on the Novel dataset. All the experiments on the Novel dataset follow an independent test set cross-validation approach.

6.3.5.4 Ablation study of various feature sets with state-of-art classifiers

As discussed in section 6.3.1, two types of features are extracted - evolutionary profile-based and embedded-based features, and these features are quantitatively analyzed in this study. Initially, a single-label classification on various feature sets is conducted on the Benchmark dataset. A single-label data is derived from multi-labeled data by transforming each data point 'x', which has k labels to k data points with its unique labels. As discussed in section 6.1, the Benchmark dataset exhibits 556 single-label data points, 21 two-label data points are transformed into 42 single-label data points, and one three-label data point is transformed into three data points. Thus, transformed single-label data for Benchmark contains 601 data points.

A. Evolutionary Profile Features:

Evolutionary profiles - PSSM (size=$L \times 42$) and HMM ($L \times 30$) matrix are extracted from a protein sequence, as mentioned in section 2.1.1. Mono-gram (size=20) features and bi-gram (400) features are extracted from the PSSM-lo ($L \times 20$) and PSSM-ls ($L \times 20$) matrix and the HMM matrix ($L \times 20$). Corresponding features of PSSM and HMM profiles are combined and are referred to as PSSM + HMM mono-gram (40) and bi-gram (800) features.

Table 6.2. The Performance (Accuracy in percentage) Analysis of Evolutionary Feature sets using various State-of-art Classifiers.

Feature Set (Size)	Classifiers						
	NB	SVM	LR	GBM	MLP	k-NN	RF
PSSM-lo mono-gram (20)	26.10	**66.44**	60.93	57.03	65.41	59.07	61.34
PSSM-ls mono-gram (20)	24.80	51.70	58.70	60.90	63.70	61.30	**67.50**
PSSM-lo bi-gram (400)	29.09	**72.21**	70.41	62.59	64.25	62.76	65.60
PSSM-ls bi-gram (400)	34.59	68.70	67.15	61.92	66.30	62.36	**68.88**
HMM mono-gram (20)	22.90	50.20	60.00	60.20	63.20	59.70	**66.50**
HMM bi-gram (400)	27.80	66.70	64.05	62.37	63.30	61.80	**66.80**
PSSM-lo + HMM mono-gram (40)	21.91	**69.88**	62.07	60.53	63.91	59.92	66.84
PSSM-lo + HMM bi-gram (800)	32.04	**69.89**	68.90	65.20	61.49	59.42	69.68

The performance of various sets of evolutionary-based profiles of mono-gram and bi-gram techniques with seven state-of-the-art classifiers are tabulated in Table 6.2. To identify the best performing features and base classifier, the results are recorded on the transformed single-labeled Benchmark dataset. The bold value indicates the best results obtained by a classifier for the given feature set. From Table 6.2, it is observed that the PSSM profile features are more effective compared to features from HMM profiles. Moreover, bi-gram features on PSSM-lo exhibit higher information when compared to bi-gram features on the PSSM-ls matrix and even better than the combined PSSM + HMM profile features. Both SVM and RF perform equally well on the Benchmark dataset. However, SVM performs consistently better and outperforms RF on the best-performing feature sets, i.e., bi-gram features of PSSM-lo.

B. Embedded Features:

As discussed in section 6.3.1.2, Word2Vec technique is adopted and modified to extract the character embedding features of size 800 in which 400 features are from the amino acid sequence (referred to as ‘Seq’ in Table 6.3) and the rest 400 features from the secondary structural sequence (referred to as ‘Str’ in Table 6.3). These combined features of size 800 are referred to as ‘Combined’ in Table 6.3. Further, various window size (ws) of Word2Vec are explored (i.e., ws from one to four), and the obtained results are tabulated in Table 6.3.

The performance of various sets of embedding-based features from protein sequence and structural sequence with seven state-of-the-art classifiers are tabulated in Table 6.3. In order to identify the best performing features and base classifier, the results are captured on the transformed single-labeled Benchmark dataset. The bold value indicates the best results obtained for a given window size

From Table 6.3, it is evident that the combined embedded features outperform either of 'Seq' or 'Str' features in all the window sizes (ws). Specifically, combined features of ws=4 have the highest information than others. Also, SVM performs the best among all state-of-the-art classifiers in classifying the sub-chloroplast location.

Table 6.3. The Performance (Accuracy in percentage) Analysis of Embedding Feature sets using various State-of-art Classifiers.

Approach	On (Size)	NB	SVM	LR	GBM	MLP	k-NN	RF
W2V ws[†]=1	Seq (400)	35.46	53.71	52.54	52.79	46.41	48.69	51.07
	Str (400)	37.28	48.70	42.17	46.83	47.40	43.25	41.07
	Combined (800)	41.64	**57.72**	54.03	52.05	48.80	52.38	55.59
W2V ws[†]=2	Seq (400)	36.81	55.41	52.87	52.73	47.03	46.46	51.40
	Str (400)	36.61	48.38	42.17	46.68	48.75	43.27	41.41
	Combined (800)	42.30	**59.07**	52.30	55.36	49.98	54.92	57.55
W2V ws[†]=3	Seq (400)	34.51	55.23	52.70	53.17	46.75	51.54	52.71
	Str (400)	37.75	48.54	42.17	46.52	46.28	42.61	41.39
	Combined (800)	39.44	**57.38**	52.69	55.18	49.74	54.24	53.89
W2V ws[†]=4	Seq (400)	35.80	57.77	52.53	54.55	51.74	50.26	55.20
	Str (400)	36.74	47.40	42.17	46.00	48.56	42.11	41.58
	Combined (800)	40.94	**60.73**	53.82	56.43	50.34	56.76	57.22

[†] window size

The outcome of this ablation study is that among various feature sets and state-of-the-art classifiers that are explored, bi-gram features on PSSM-lo is the best feature set, and SVM is the best classifier. Hence, for further comparative analysis, the PSSM-lo feature set and SVM combination are chosen for the multi-label BR framework.

6.3.5.5 Multi-label classification: GA-BiSVM vs. BiSVM

From the previous section 6.3.5.4 outcome, it is observed that PSSM-lo bi-gram features exhibit higher discriminating information of sub-chloroplast locations for the single-label Benchmark dataset, and these features are utilized and classified effectively by the SVM classifier. Therefore, the PSSM-lo bi-gram (400) and SVM are shortlisted for multi-label study.

For multi-label prediction, two models - BiSVM and GA-BiSVM are proposed (as discussed in section 6.3.4). Both models are evaluated using the Leave-one-out cross-validation (LOOCV) statistical approach on the multi-label Benchmark dataset (to find the effect of label-specific feature selection using GA on multi-label classification). The prediction results are tabulated in Table 6.4.

Table 6.4. The Performance Comparison (in percentage) of GA-BiSVM and BiSVM on multi-label Benchmark Dataset using LOOCV

Evaluation Measure	GA-BiSVM	BiSVM
OAA	64.19	**66.67**
ACC	67.76	**68.97**
Precision	68.60	**69.64**
Recall	**70.73**	**70.73**
F1 score	69.00	**69.78**
Grand Mean	68.06	**69.15**

The proposed BiSVM model performed consistently better than the proposed GA-BiSVM model by 1 to 1.5% in all measures. This is due to of two main reasons: Firstly, the dataset was highly imbalanced, i.e., data objects of some labels are very less, when compared to other labels. In the Benchmark dataset, Lumen label and plastoglobule constitute only 5% each from the total dataset, resulting in more negative classes in the label-specific model. Thus, these label-specific features are highly biased for negative class prediction. Secondly, all the 400 PSSM features are important and relevant, and any further reduction of features degrades the multi-label prediction accuracy.

6.3.5.6 Comparison of the proposed model with existing State-of-art-predictors on Benchmark dataset

The proposed BiSVM model performance on the Benchmark dataset is compared with three state-of-the-art-predictors, i.e., AL-KNN Lin *et al.* (2013), MultiP-SChlo Wang *et al.* (2015), and EnTrans-Chlo Wan *et al.* (2016*b*). All the results of evaluation measures obtained from respective models using LOOCV are tabulated in Table 6.5.

AL-KNN and MultiP-SChlo utilize Pseudo amino acid composition features (PseAAC), whereas EnTrans-Chlo adopts an ensemble of PseAAC and PSSM based features. MultiP-SChlo performs multi-label classification using GA selected features and Binary Relevance with SVM as the base classifier. Both AL-KNN and EnTrans-Chlo adopt the adaptive approach. AL-KNN utilizes ML-KNN, whereas EnTrans-Chlo takes a transductive approach with the least-squares as an error function and k-NN algorithm.

The proposed BiSVM model performs significantly better than MultiP-SChlo in all

Table 6.5. The Performance Comparison (in percentage) of the Proposed BiSVM Model with state-of-art predictors on Benchmark dataset using LOOCV

Evaluation Measure	AL-KNN	MultiP-SChlo	EnTrans-Chlo	BiSVM
OAA	43.77	55.52	60.03	**66.67**
ACC	45.21	63.26	66.00	**68.97**
Precision	46.63	64.10	67.30	**69.64**
Recall	45.30	**71.06**	**71.06**	70.73
F1 Score	45.95	67.38	68.04	**69.78**
Grand Mean	45.37	64.26	66.49	**69.15**

performance metrics except for the recall, even as the proposed BiSVM and the MultP-SChlo model utilizes a similar classification approach. The main difference in performance is due to the extraction of discriminating features by the proposed approach, i.e., PSSM-lo bi-gram features exhibit higher information compared to the PseAAC. The BiSVM achieves significantly better performance in the Overall Actual Accuracy (OAA) metric, i.e., 11% higher than MultiP-SChlo. The BiSVM performs better than EnTrans-Chlo by 6% in OAA and other performance metrics by 2% except for recall. The transductive model is unable to take advantage of the test data points in training the model using LOOCV (as there is only one test case). Thus, BiSVM operating on similar evolutionary features, i.e., PSSM features, performs better than EnTrans-Chlo.

Validation on Unknown dataset: Novel Dataset:

We observe that the performance of the proposed model has been effective on the Benchmark dataset when compared to other state-of-the-art models. To validate the generalization of the proposed model, a comparative study is carried out on an unknown dataset, i.e., Novel dataset. The Novel dataset is a derived dataset (as mentioned in section 6.1). It contains the same labels as the Benchmark dataset except for the plastoglobule label.

Table 6.6. The Performance Comparison (in percentage) of the Proposed BiSVM Model with state-of-art predictors on Unknown Test Set, i.e., Novel dataset

Evaluation measure	MultiP-SChlo	EnTrans-Chlo	BiSVM
OAA	27.05	36.07	**47.54**
ACC	32.79	46.31	**54.37**
Precision	35.25	48.50	**56.15**
Recall	36.07	54.92	**59.43**
F1 score	34.70	49.86	**56.69**
Grand Mean	33.17	47.13	**54.83**

Table 6.6 compares the performance of BiSVM with the other two existing state-of-the-art models. All the three models mentioned in Table 6.6 are trained on the Benchmark dataset and tested on the Novel dataset. BiSVM outperforms the other two state-of-the-art predictors in classifying the unknown dataset. BiSVM performs 20-23% and 7% better than MultiP-SChlo and EnTrans-Chlo predictor, respectively, across the performance metrics.

6.3.5.7 Statistical Analysis

The proposed BiSVM model outperformed all the state-of-the-art models on OAA by a minimum margin of 6.64% to the maximum margin of 22.90% on the Benchmark dataset. To demonstrate the improvement significance of the results, we have performed a statistical paired t-test on the PSCL OAA metric among the proposed BiSVM with the other three models from literature such as AL-KNN Lin *et al.* (2013), MultiP-SChlo Wang *et al.* (2015), and EnTrans-Chlo Wan *et al.* (2016*b*).

Let a null hypothesis indicate that there is no significant difference between the proposed BiSVM with the other three models with a significance level of 5% (i.e., 0.05). When $p < 0.05$, the null hypothesis is rejected, and it indicates that there is indeed a statistically significant difference in the results. Otherwise, i.e., when $p > 0.05$, the null hypothesis is retained, and it indicates that there is no significant difference in the results. The results of the paired t-test are shown in Table 6.7.

Table 6.7. The Paired t-test among the Proposed BiSVM and other three State-of-the-art Models on Benchmark dataset

Methods	**p-value**	**Null Hypothesis Decision**	**Significant Difference**
AL-KNN Lin *et al.* (2013)	<0.00001	Reject	Yes
MultiP-SChlo Wang *et al.* (2015)	0.000184	Reject	Yes
EnTrans-Chlo Wan *et al.* (2016*b*)	0.000956	Reject	Yes

From Table 6.7, we observe that the proposed BiSVM model rejected the null hypothesis against all the state-of-the-art models on the Benchmark dataset. Hence, we claim that the proposed BiSVM model is effective in solving the multi-label PSCL problem.

The main outcome of the proposed BiSVM model are as follows:

- Character embedding features were not as effective when compared to evolutionary-based features.

- Bigram feature extraction technique was effective when compared to mono-gram feature extraction technique as the bi-gram technique was able to find more discriminating patterns over mono-gram technique.
- PSSM-based features were effective when compared to HMM-based features.
- GA feature selection on a combined superset of all features was proved to be ineffective when compared to only PSSM features.

The performance of the proposed model of this preliminary study is further improved by enhancing feature modeling and with the deep learning framework. This will be discussed in the next section 6.4 in detail.

6.4 Protein Sub-Chloroplast Localization Prediction using Deep Neural Network

Based on the previous investigation on multi-label PSCL prediction, i.e., the BiSVM model reported satisfactory performance. However, the effectiveness of the PSSM bigram features BR framework on both datasets was limited. There is a scope to explore the SXG feature extraction technique to improve the performance. Moreover, the BR framework is sensitive to the class imbalance problem. Therefore, to address the class imbalance limitation, a deep learning framework has been proposed.

6.4.1 Feature Extraction

A quality set of features play an important role in solving the PSCL prediction problem. In this study, we propose an effective evolutionary-based feature extraction technique named SkipXGram bi-gram (SXGbg).

6.4.1.1 SkipXGram bi-gram (SXGbg) Technique

The local interactions of amino acid residues in a protein sequence play an important role in identifying its locations. A bigram is one of the well-known and effective techniques to extract the local interactions of amino acid residues in close proximity Sharma *et al.* (2013*a*); Lyons *et al.* (2015). The bigram technique extracts 400 feature vectors from a protein sequence as the protein sequences are made up of 20 different amino acids (i.e., 20×20).

In the previous chapter 5.2.1.2 SXGbg feature extraction technique was successfully explored and analyzed on fold prediction. Since it was proven to be an effective approach, in this study, to extract important and various levels of local interactions, we have adopted the SXGbg to extract feature sets from evolutionary-based profiles. To the best of our knowledge, this is the first work to explore a profile-based skip-gram technique to predict the protein sub-chloroplast location.

Seven levels of amino acid local interactions are captured from an evolutionary profile using equation 6.9 by skipping zero to six consecutive grams (residues). The skip level is denoted as X and the values vary from zero to six.

$$SXGbg(i,\ j) = \sum_{l=1,\ X\epsilon\{0-6\}}^{L-X-1} EP_{(l,\ i)} \times EP_{(l+X+1,\ j)} \tag{6.9}$$

where, $1 \leq i, j \geq 20$, EP is an evolutionary-based profile of a dimension $L \times 20$ and L is a length of a query sequence. A seven sets of features are extracted of which each set of size 400.

6.4.2 Multi-label Classification

6.4.2.1 Deep Neural Network

The features that are extracted from evolutionary profiles using the proposed SXGbg technique are fed into the proposed fully connected deep feed-forward neural network and it is shown in Figure 6.2. The proposed deep neural network consists of three fully connected hidden layers, followed by an output layer and a concatenate layer to predict the multi-label protein sub-chloroplast locations. In this study, two evolutionary-based profiles are explored, i.e., HMM and PSSM. Seven sets of features are extracted from each evolutionary-based profile. A total of 14 sets of features (each set consisting of 400 features) are analyzed individually and the best performing feature set is shortlisted. The detailed analysis is available in section 6.4.3.

Hidden Layers: The extracted features (of size 400) from SXGbg are fed into the first hidden layer consisting of 400 neurons. The output of the first hidden layer is fed into the second hidden layer, which consists of 100 neurons. The output of the second hidden layer is fed into the third hidden layer, which consists of 25 neurons. All the neurons of three hidden layers are activated by a sigmoid function using the equation 6.10. The sigmoid function gradually transforms a wide range of input values to real numbers in an interval of 0 to 1. The main advantage of the sigmoid function is that it is simple and achieves better training performance for multi-layer neural networks via back-propagation as the output range is between 0 to 1 Karlik and Olgac (2011).

$$sigmoid(x) = \left(\frac{1}{1+e^{-x}}\right) \tag{6.10}$$

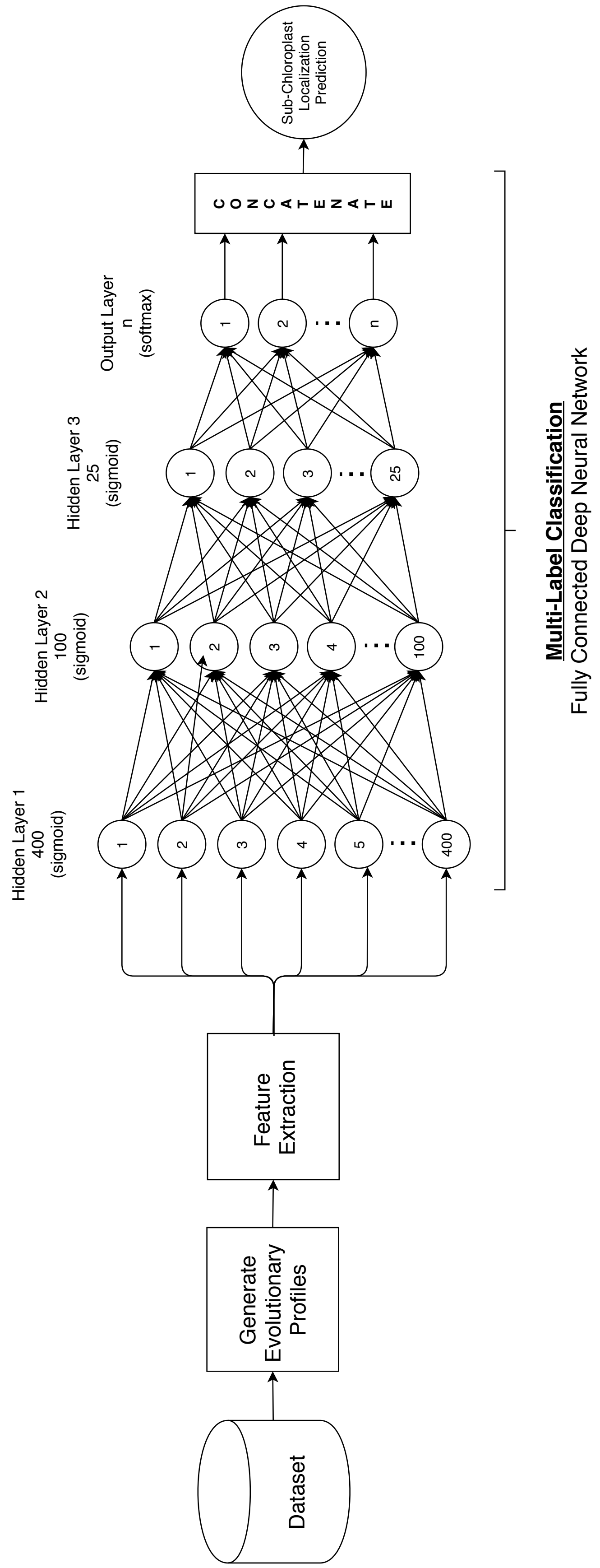


Figure 6.2. The Proposed Framework for PSCL consisting of Feature Extraction with Fully Connected Deep Neural Network.

Output Layer: The output layer consists of n neurons, where n is the distinct number of sub-chloroplast locations and all the neurons are activated by softmax function. The output of the last hidden layer is fed into the output layer to predict the location probabilities for a given query protein sequence. The output of the softmax function Goodfellow *et al.* (2016) is computed by equation 6.11, and the output values of the output layer are in the range of 0 to 1.

$$softmax\left(x_i\right) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{6.11}$$

Concatenate Layer: The label probabilities of the output layer are rounded-off (i.e., value $\leq$ 0.5 is treated as 0 and the value $>$ 0.5 is treated as 1) and concatenated to obtain the final multi-label prediction of a query sequence.

6.4.2.2 Hyper-parameter Optimization

A stochastic gradient descent algorithm Kingma and Ba (2014) algorithm is adopted to calculate the training error of the proposed deep neural network and the weights are updated via backpropagation. The primary goal of the training model is to minimize the cross-entropy loss function, as shown in equation 6.12, where Loc is the number of sub-locations of chloroplast proteins, *log* is the natural logarithmic function, y_i is the actual locations for the i^{th} protein sequence, p_i is the predicted locations for the same protein sequence, and γ is the L2 regularization hyper-parameter. On the other hand, the optimization of all the parameters is performed according to equation 6.13, where ϕ is the parameter rate and β is the learning rate.

$$L\left(\phi\right) = -\sum_{i=1}^{Loc} y_i \left(\log\left(p_i\right)\right) + \gamma \|\phi\|^2 \tag{6.12}$$

$$\phi \leftarrow \phi + \beta \frac{\partial L\left(\phi\right)}{\partial \phi} \tag{6.13}$$

The number of hidden layers is varied from 1 to 10 and the best results are reported for three hidden layers. The number of neurons in hidden layers are explored with various sizes from 1024 to 8, and the best-obtained values are 400, 100, and 25 neurons for the first, second, and third hidden layer, respectively. The best combination of parameter values as mentioned above are identified using grid-search, and these values are kept constant on all the experiments that are discussed in the next section.

6.4.3 Results and Discussion

The performance of the proposed model is discussed in three stages: Initially, an ablation study is performed on various sets of extracted features from evolutionary profiles on both datasets. Next, the best performing feature set is shortlisted and compared with the state-of-the-art models. Finally, the significance of performance improvement is verified by conducting a statistically significant test.

6.4.3.1 Experiment Setup

The proposed SXGbg feature extraction technique is implemented in Python 3 and the proposed multi-label deep neural network is implemented with the support of Keras libraries (provided by Tensorflow) Abadi *et al.* (2016). All the experiments of the proposed model are carried out on an Ubuntu-based server having 128 GB RAM, 56 cores of Intel Xeon processors, two NVIDIA Tesla M40 GPUs, and a 3TB hard drive.

6.4.3.2 Evaluation Metrics

We have followed the same evaluation metrics that were discussed in section 6.3.5.2.

6.4.3.3 Cross Validation

In this study, we have followed the same cross-validation approach for both the datasets that were discussed in the section 6.3.5.3

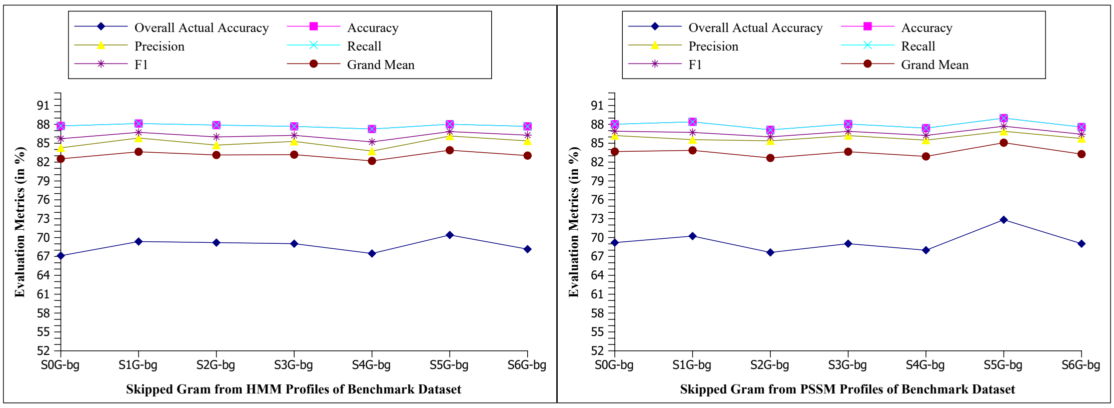
6.4.3.4 An Ablation Study on Evolutionary Profiles-based Features

Seven sets of skipped gram features are extracted from each evolutionary-based profile by varying the skip value (i.e., X from 0 to 6) using the proposed SXGbg technique. A total of 14 SXGbg feature sets are extracted for a given dataset in which seven sets are from HMM profiles and the other seven are from PSSM profiles. The protein sub-chloroplast prediction performances of each skipped gram feature set for the Benchmark dataset is tabulated in Table 6.8. A similar study is performed on the Novel dataset and the results are tabulated in Table 6.9. A bold value in Table 6.8 and 6.9 represent the best result obtained for the respective SXG feature set.

For the Benchmark dataset, two important things are observed from the Table 6.8 and those are: (i) the S5Xbg feature set reported higher performance for all the evaluation metrics irrespective of evolutionary profiles and the same can be observed in Figure 6.3. The multi-label prediction accuracy, i.e., OAA of S5Gbg outperformed other skipped gram feature sets by a minimum margin of 1% on HMM and 2.5% on PSSM profile; (ii) the SXGbg feature sets of PSSM profile reported higher prediction performance on all the evaluation metrics when compared to SXGbg feature sets of HMM profile.

Table 6.8. The Performance (in percentage) of the Proposed Model on Various SXGbg Feature Sets Extracted from Evolutionary Profile of Benchmark Dataset.

Evolutionary Profile	**Evaluation Metrics**	**Skipped-gram Feature Sets' Results (in %)**						
		S0G	**S1G**	**S2G**	**S3G**	**S4G**	**S5G**	**S6G**
HMM	OAA	67.12	69.37	69.20	69.03	67.47	**70.41**	68.16
	Accuracy	87.75	88.13	87.88	87.68	87.26	**88.02**	87.68
	Precision	84.27	85.83	84.70	85.29	83.75	**86.15**	85.37
	Recall	87.75	88.13	87.88	87.68	87.26	**88.02**	87.68
	F1	85.72	86.73	86.00	86.25	85.21	**86.85**	86.28
	Grand Mean	82.52	83.64	83.13	83.19	82.19	**83.89**	83.03
PSSM	OAA	69.20	70.24	67.64	69.03	67.99	**72.83**	69.03
	Accuracy	88.02	88.40	87.13	88.06	87.40	**88.99**	87.57
	Precision	86..22	85.57	85.38	86.21	85.49	**86.90**	85.75
	Recall	88.02	88.40	87.12	88.06	87.40	**88.99**	87.57
	F1	86.91	86.72	86.01	86.89	86.24	**87.70**	86.43
	Grand Mean	83.67	83.87	82.66	83.65	82.90	**85.08**	83.27



(a) Features from HMM Profile

(b) Features from PSSM Profile

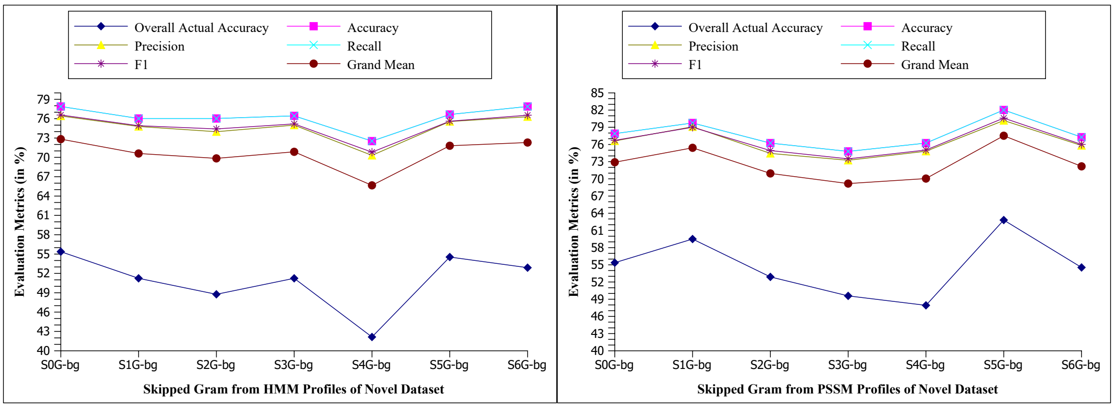
Figure 6.3. The Performance Prediction of Various Skipped Gram Features on Benchmark Dataset

Also, similar observations as above hold good for the Novel dataset and the same is evident from Table 6.9 and Figure 6.4. Further, it is worth noting that the prediction performance of the S5Gbg feature set reported higher OAA prediction performance across other feature sets. The prediction performance of the S5Gbg PSSM feature set outperformed in all evaluation metrics when compared to HMM by a margin of 4% to 8%.

Discussion: From the Figure 6.5, it is evident that the OAA and Grad Mean evaluation metrics of the S5Gbg feature set reported higher prediction performance across other SXGbg feature sets. This is due to the fact that five skipped grams of bigrams

Table 6.9. The Performance (in percentage) of the Proposed Model on Various Skipped-gram Feature Sets Extracted from Evolutionary Profile of Novel Dataset.

Evolutionary Profile	**Evaluation Metrics**	**Skipped-gram Feature Sets' Results (in %)**						
		S0G	**S1G**	**S2G**	**S3G**	**S4G**	**S5G**	**S6G**
HMM	OAA	55.37	51.24	48.76	51.24	42.14	**54.54**	52.89
	Accuracy	77.89	76.03	76.03	76.44	72.52	76.65	**77.89**
	Precision	76.42	74.79	73.98	75.00	70.31	75.56	**76.3**
	Recall	77.89	76.03	76.03	76.44	72.52	76.65	**77.89**
	F1	76.57	74.91	74.41	75.19	70.81	75.61	**76.54**
	Grand Mean	72.83	70.60	69.84	70.86	65.66	71.80	**72.30**
PSSM	OAA	55.37	59.50	52.89	49.58	47.93	**62.81**	54.54
	Accuracy	77.89	79.75	76.24	74.79	76.24	**82.02**	77.27
	Precision	76.62	79.09	74.44	73.24	74.82	**80.18**	75.82
	Recall	77.89	79.75	76.23	74.79	76.24	**82.02**	77.27
	F1	76.72	79.00	74.94	73.49	75.03	**80.60**	76.01
	Grand Mean	72.90	75.42	70.95	69.18	70.05	**77.53**	72.18



(a) Features from HMM Profile

(b) Features from PSSM Profile

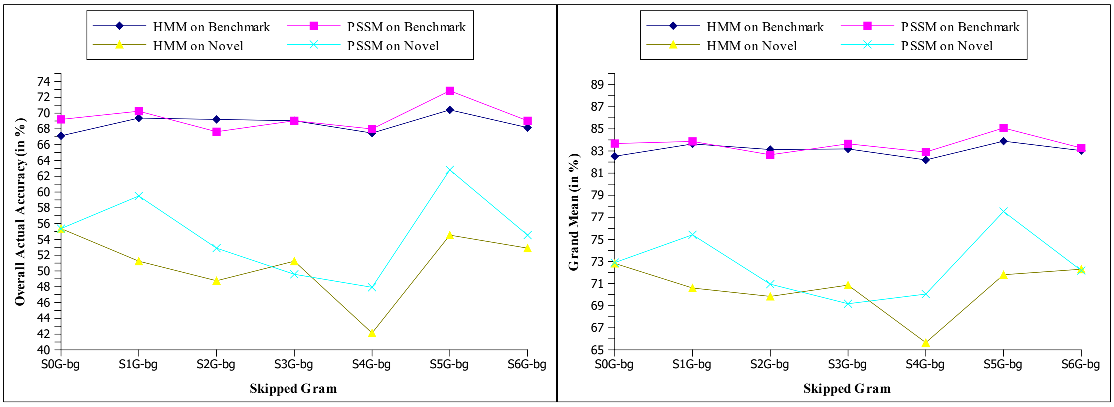
Figure 6.4. The Performance Prediction of Various Skipped Gram Features on Novel Dataset

carry rich evolutionary information of a possible protein structure that is mainly due to various local interactions; thus, able to capture higher location patterns.

From the Figure 6.6, it is evident that the S5Gbg feature set of the PSSM profile outperforms the S5Gbg feature set of HMM profile across both the datasets. This is due to the fact that PSSM profiles are known to extract high homologous evolutionary information when compared to HMM profiles. Thus, only S5Gbg features of PSSM (of size 400) are considered for the rest of the experiment analysis.

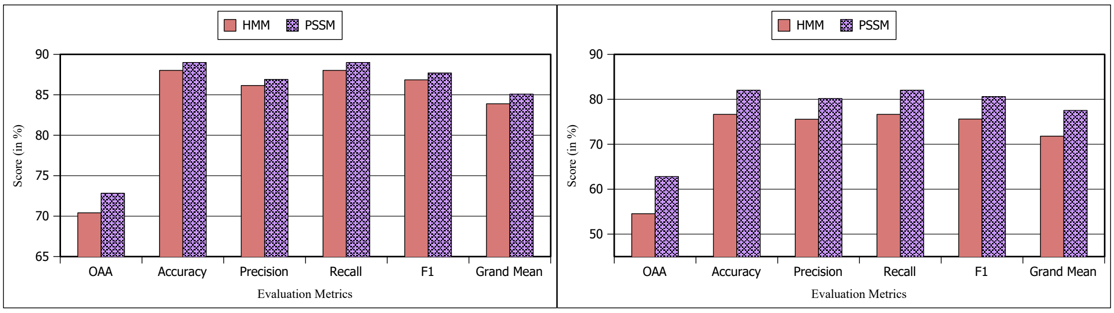
6.4.3.5 Comparison with state-of-the-art models

To demonstrate the effectiveness of the proposed model, the performance of all the evaluation metrics are compared with the state-of-the-art models. The PSCL predic-



(a) Overall Actual Accuray Metric (b) Grand Mean Metric

Figure 6.5. The Performance Comparison of Evolutionary Profiles across Various Skipped Gram Features



(a) on Benchmark Dataset (b) on Novel Dataset

Figure 6.6. The Performance Comparison of S5Gbg features of Evolutionary Profiles across Various Evaluation Metrics

tion performance of the state-of-the-art models and the proposed model on Benchmark and Novel datasets are tabulated in Table 6.10. The performance comparison of the proposed model with state-of-the-art models is shown in Figure 6.7.

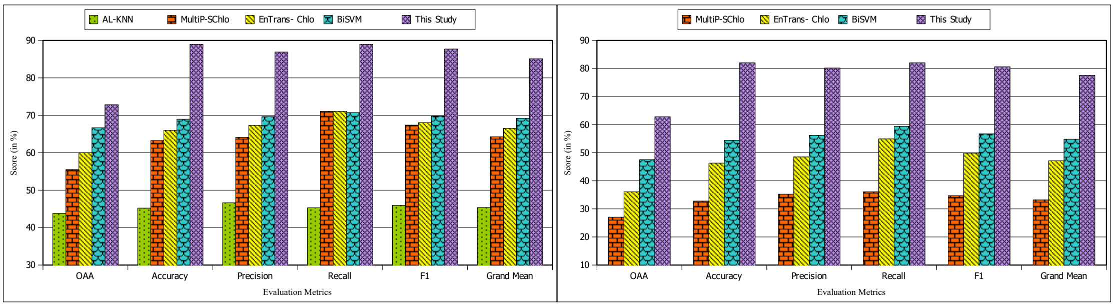
The experiments from all the state-of-the-art models including this study on the Benchmark dataset are carried out using leave-one-out cross-validation. Whereas the experiments from all the state-of-the-art models including this study on the Novel dataset are carried-out using independent test cross-validation, i.e., the model is trained on a Benchmark dataset and tested on the Novel dataset. The state-of-the-art models' results on both the datasets are taken from their published work and (-) indicates the unavailability of the result.

The multi-label accuracy (i.e., OAA metric) of the proposed model recorded 72.83% and 62.81% on Benchmark and Novel dataset respectively. The overall (Grand Mean metric) prediction performance of the proposed model recorded 85.08% and 77.52% on Benchmark and Novel dataset respectively.

From Table 6.10, it is evident that the proposed model outperformed all the multi-

Table 6.10. The Performance Comparison (in percentage) of the Proposed Model against the State-of-the-art Models on Benchmark and Novel Datasets.

Dataset	**Evaluation Metrics**	**AL-KNN**	**MultiP-SChlo**	**EnTrans-Chlo**	**BiSVM**	**This Study**
Benchmark	OAA	43.77	55.52	60.03	66.67	**72.83**
	Accuracy	45.21	63.26	66.00	68.97	**88.99**
	Precision	46.63	64.10	67.30	69.94	**86.90**
	Recall	45.30	71.06	71.06	70.73	**88.99**
	F1	45.95	67.38	68.04	69.78	**87.70**
	Grand Mean	45.37	64.26	66.49	69.15	**85.08**
Novel	OAA	-	27.05	36.07	47.54	**62.81**
	Accuracy	-	32.79	46.31	54.37	**82.02**
	Precision	-	35.25	48.50	56.15	**80.18**
	Recall	-	36.07	54.92	59.43	**82.02**
	F1	-	34.7	49.86	56.69	**80.60**
	Grand Mean	-	33.17	47.13	54.83	**77.52**



(a) on Benchmark Dataset

(b) on Novel Dataset

Figure 6.7. The Performance Comparison (in percentage) of the Proposed Model against the State-of-the-art models

label state-of-the-art models across both the datasets. The Grand Mean performance of the proposed model has been enhanced absolutely by a minimum margin of 15.93% and 22.69% on Benchmark and Novel datasets respectively when compared to the next best model i.e., BiSVM (as discussed in section 6.3).

Discussion: From Table 6.10, it can be observed that state-of-the-art models such as, MultiP-SChlo Wang *et al.* (2015), EnTrans-Chlo Wan *et al.* (2016*b*) and BiSVM reported a Grand Mean of 64.26%, 66.49%, and 69.15% respectively on Benchmark dataset; whereas, the Grand Mean performance reduced to 33.17%, 47.13%, and 54.83% respectively on Novel dataset.

It is worth noting that the Grand Mean performance of state-of-the-art models such as MultiP-SChlo Wang *et al.* (2015), EnTrans-Chlo Wan *et al.* (2016*b*) and BiSVM reduced relatively by 48.2%, 29.1%, and 20.70% respectively on Novel dataset when compared to their performances on the Benchmark dataset. However, the proposed model recorded a Grand Mean of 85.08% and 77.52% on Benchmark and Novel dataset respectively and the proposed model's Grand Mean performance on Novel dataset is reduced relatively by only 8.8% when compared to its performance on Benchmark dataset. Hence, it is acceptable to claim that the proposed model is not only effective but also it is a more generalized model in predicting protein sub-chloroplast localization when compared to other state-of-the-art models.

6.4.3.6 Statistical Significance Analysis

To demonstrate the significant improvement in the Overall Actual Accuracy (OAA) of the proposed model, we have carried out a statistical paired t-test on the OAA among the proposed model with the other state-of-the-art models from the literature. The significance test for AL-KNN Lin *et al.* (2013) is performed only with respect to Benchmark dataset results due to unavailability of results on Novel dataset.

Let a null hypothesis indicate that there is no significant difference between the proposed model and the other state-of-the-art models with a significance level of 5% (i.e., 0.05). When $p < 0.05$, the null hypothesis is rejected, and it indicates that there is indeed a statistically significant difference in the results. Otherwise, i.e., when $p > 0.05$, the null hypothesis is retained, and it indicates that there is no significant difference in the results. The results of the paired t-test are shown in Table 6.11.

Table 6.11. The Statistical Paired t-test between the Proposed Model and the State-ofthe-art Model on Overall Actual Accuracy

Model	Dataset	p-value	Hypothesis Decision	Significant Difference
AL-KNN Lin *et al.* (2013)	Benchmark	<.00001	Reject	Yes
MultiP-SChlo Wang *et al.* (2015)	Benchmark	<.00001	Reject	Yes
	Novel	<.00001	Reject	Yes
EnTrans-Chlo Wan *et al.* (2016*b*)	Benchmark	<.00001	Reject	Yes
	Novel	<.00001	Reject	Yes
BiSVM	Benchmark	<.00001	Reject	Yes
	Novel	<.00001	Reject	Yes

From Table 6.11, it is observed that the proposed model rejected the null hypothesis on all the state-of-the-art models across both the datasets. Hence, we claim that the proposed model is effective in solving the protein sub-chloroplast localization prediction.

6.5 Summary

Identification of proteins that are located in the sub-chloroplast compartments help in further understanding their roles in the various chloroplast biological activities. The PSCL prediction is a multi-label problem. This chapter elaborated two proposed models for the multi-label PSCL prediction problem. In the first model (i.e., BiSVM), PSSM-based bi-gram features proved to be effective with the BR framework to solve PSCL. In the next model, the limitations of BiSVM are solved by SXGbg features with multi-label deep learning architecture. The later model outperformed (in Grand Mean metric) all the state-of-the-art models by a minimum margin of an absolute 15.93% and 22.69% on Benchmark and Novel datasets respectively. Statistical significance test on the performance improvement shows that the prediction of the proposed model is effective in the identification of the PSCL.

The next chapter concludes our thesis with a summary of the work done and presents some suggestions for future work in this area.

Chapter 7

Conclusion and Future Work

The main objective of this thesis was to propose effective computational models that help in the identification of protein structure and its subcellular localization. This thesis achieved all the main objectives by proposing effective models for multiple sequence alignment, protein secondary structural class prediction, protein fold recognition, and protein subcellular localization prediction.

7.1 Conclusion

An effective and computationally feasible (polynomial time) alignment model was proposed with a novel scoring system and optimization framework. The proposed scoring system incorporated two effective strategies, i.e., LBA and PRSDGP in which the LBA scoring strategy calculates the score of a current residue pair based on the previous position status information, and the PRSDGP scoring strategy dynamically calculates the gap penalty value based on its position and residue information using the mutation matrix. The proposed SIO framework identifies and optimizes the aligned results using the proposed scoring strategies to overcome the local optima limitation of the progressive approach. The proposed ProgSIO-MSA model being a progressive approach was evaluated against both progressive and iterative-based models on benchmark datasets. The experimental results showed that the accuracy (quality) of the proposed ProgSIO-MSA model, when compared with the CLUSTAL X model (best state-of-the-art progressive model), was increased by 17.7% on the BAliBASE dataset. The proposed ProgSIO-MSA model performance was equally good when compared to the best stochastic-based iterative model, i.e., GAPAM. Moreover, the computational efficiency of the proposed ProgSIO-MSA model outperformed the CLUSTAL model in running time and outperformed GAPAM in time complexity by a factor of $[G \,.\, P]$ (for G number of generations and P number of populations). It was also observed that the performance improvement of the proposed model was statistically significant.

An effective and generic computational model was proposed to predict the PSSC effectively for both high and low similarity datasets. The proposed model consists of an enhanced feature modeling with an ensemble of three classifiers. The proposed feature modeling consists of three feature extraction techniques such as Character Embedding (CE), SkipXGram (SXGbg), and General Statistical (GS) based feature extraction technique. The proposed model reported 93.55% and 97.58% overall accuracy for high similarity datasets namely, z277 and z498 respectively. For low sequence similarity datasets, the proposed model attained 81.82%, 81.12%, and 93.93% on 25PDB, 1189, and FC699 datasets respectively. The performance of the proposed model reported the highest overall accuracy across various benchmark datasets and outperformed all the state-of-the-art models for both low and high similarity datasets. Further, the assessment of the proposed model on the updated high-volume dataset, i.e., SCOPe_2.07 showed that the performance of the proposed model is consistent and robust even for the large-scale updated dataset. It was also observed that the performance improvement of the proposed model was statistically significant.

An effective computational model was designed and developed to solve protein fold recognition. The proposed model consists of a novel combination of feature extraction techniques such as Convolutional (Conv) and SkipXGram bi-gram (SXGbg) and the fold recognition was performed using the proposed deep learning architecture. The performance of the proposed model was assessed on three benchmarks and the latest derived high-volume datasets. The proposed model reported 85.9%, 95.8%, and 88.8% on DD, EDD, and TG benchmark datasets respectively. The performance of the proposed model improved by 5% to 23%, 2% to 19%, and 3% to 30% on DD, EDD, and TG datasets, respectively when compared to the best models from the literature. The performance of the proposed model recorded 91.4% on one of the derived high-volume datasets. It was also observed that the performance improvement of the proposed model was statistically significant.

An effective computational model was designed and developed to solve multi-label protein sub-chloroplast localization prediction. The proposed model consists of a novel and effective SXGbg feature extraction technique and the multi-label location prediction was performed using the proposed deep learning architecture. The performance of the proposed model was assessed on two benchmark datasets. The proposed model outperformed (in Grand Mean metric) all the state-of-the-art models by a minimum margin of an absolute 15.93% and 22.69% on Benchmark and Novel datasets respectively. It was also observed that the performance improvement of the proposed model was statistically significant.

7.2 Future Work

The work discussed in this thesis has inspired a couple of promising directions for future research and they are outlined below:

- In this research work, we adopted PSSM and HMM evolutionary-based profiles for predicting PFR and PSCL problems. The proposed ProgSIO-MSA alignment model can be further extended to search the closely related amino acid sequences from the NR database and a novel evolutionary-based profile can be generated using the proposed ProgSIO-MSA alignment model.

- To analyze and validate the effectiveness of novel evolutionary-based profiles on the proposed models of PSSC prediction problem, PFR problem, and PSCL prediction problem.

- To explore feature selection and optimization techniques to solve the prediction of PSSC and PFR problems. In PSSC prediction, the proposed model utilizes a feature vector of size 1618 and in PFR, a proposed model utilizes a feature vector of 2512. In both the proposed models, there is a scope to apply feature selection techniques such that the reduction of feature vector size without compromising the effectiveness of the models.

- The proposed novel feature extraction techniques such as SXGbg, CE, and convolutional operations followed by deep learning architectures can be explored on further challenges of protein sequence analysis such as protein tertiary prediction, protein function prediction, and protein-protein interaction prediction.

References

Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, Tensorflow: A system for large-scale machine learning. *In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. USENIX Association, Savannah, GA, 2016. ISBN 978-1-931971-33-1. URL `https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi`.

Alberts, B., D. Bray, J. Lewis, M. Raff, K. Roberts, and J. Watson, *Molecular biology of the cell. 1994, New York and London*. Garland Publishing, Inc, 1994.

Altschul, S. F., T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17), 3389–3402.

Andreeva, A., D. Howorth, S. E. Brenner, T. J. Hubbard, C. Chothia, and A. G. Murzin (2004). Scop database in 2004: refinements integrate structure and sequence family data. *Nucleic acids research*, 32(suppl 1), D226–D229.

Andreeva, A., D. Howorth, J.-M. Chandonia, S. E. Brenner, T. J. Hubbard, C. Chothia, and A. G. Murzin (2007). Data growth and its impact on the scop database: new developments. *Nucleic acids research*, 36(suppl_1), D419–D425.

Aram, R. Z. and N. M. Charkari (2015). A two-layer classification framework for protein fold recognition. *Journal of theoretical biology*, 365, 32–39.

Bahr, A., J. D. Thompson, J.-C. Thierry, and O. Poch (2001). Balibase (benchmark alignment database): enhancements for repeats, transmembrane sequences and circular permutations. *Nucleic Acids Research*, 29(1), 323–326.

Bairoch, A. and R. Apweiler (2000). The swiss-prot protein sequence database and its supplement trembl in 2000. *Nucleic acids research*, 28(1), 45–48.

Barton, G. J. and M. J. Sternberg (1987). A strategy for the rapid multiple alignment of protein sequences: confidence levels from tertiary structure comparisons. *Journal of molecular biology*, 198(2), 327–337.

Berardi, M. J., W. M. Shih, S. C. Harrison, and J. J. Chou (2011). Mitochondrial uncoupling protein 2 structure determined by nmr molecular fragment searching. *Nature*, 476(7358), 109.

Blazewicz, J., W. Frohmberg, M. Kierzynka, and P. Wojciechowski (2013). G-msa—a gpu-based, fast and accurate algorithm for multiple sequence alignment. *Journal of Parallel and Distributed Computing*, 73(1), 32–41.

Bouchaffra, D. and J. Tan, Protein fold recognition using a structural hidden markov model. *In 18th International Conference on Pattern Recognition (ICPR'06)*volume3. IEEE, 2006.

Brady, S. and H. Shatkay, Epiloc: a (working) text-based system for predicting protein subcellular location. *In Biocomputing 2008*. World Scientific, 2008, 604–615.

Cai, Y.-D. and G.-P. Zhou (2000). Prediction of protein structural classes by neural network. *Biochimie*, 82(8), 783–785.

Carlacci, L., K. C. Chou, and G. M. Maggiora (1991). A heuristic approach to predicting the tertiary structure of bovine somatotropin. *Biochemistry*, 30(18), 4389–4398.

Chen, C., Y.-X. Tian, X.-Y. Zou, P.-X. Cai, and J.-Y. Mo (2006). Using pseudo-amino acid composition and support vector machine to predict protein structural class. *Journal of theoretical biology*, 243(3), 444–448.

Chen, K. and L. Kurgan (2007). Pfres: protein fold classification by using evolutionary information and predicted secondary structure. *Bioinformatics*, 23(21), 2843–2850.

Chen, K., L. A. Kurgan, and J. Ruan (2008). Prediction of protein structural class using novel evolutionary collocation-based sequence representation. *Journal of Computational Chemistry*, 29(10), 1596–1604.

Chen, K., W. Stach, L. Homaeian, and L. Kurgan (2011). ifc 2: an integrated web-server for improved prediction of protein structural class, fold type, and secondary structure content. *Amino Acids*, 40(3), 963–973.

Cheng, X., X. Xiao, and K.-C. Chou (2018). ploc-mgneg: Predict subcellular localization of gram-negative bacterial proteins by deep gene ontology learning via general pseaac. *Genomics*, 110(4), 231–239.

Chinnasamy, A., W.-K. Sung, and A. Mittal (2005). Protein structure and fold prediction using tree-augmented naive bayesian classifier. *Journal of Bioinformatics and computational Biology*, 3(04), 803–819.

Chothia, C. and A. V. Finkelstein (1990). The classification and origins of protein folding patterns. *Annual review of biochemistry*, 59(1), 1007–1035.

Chou, K.-C. (2000). Prediction of protein structural classes and subcellular locations. *Current Protein and Peptide Science*, 1(2), 171–208.

Chou, K.-C. (2001). Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins: Structure, Function, and Bioinformatics*, 43(3), 246–255.

Chou, K.-C. and H.-B. Shen (2007). Recent progress in protein subcellular location prediction. *Analytical biochemistry*, 370(1), 1–16.

Chou, K.-C. and H.-B. Shen (2008). Cell-ploc: a package of web servers for predicting subcellular localization of proteins in various organisms. *Nature protocols*, 3(2), 153–162.

Chou, K.-C. and H.-B. Shen (2010). Cell-ploc 2.0: an improved package of web-servers for predicting subcellular localization of proteins in various organisms. *Natural Science*, 2(10), 1090.

Chou, K.-C., D.-Q. Wei, Q.-S. Du, S. Sirois, and W.-Z. Zhong (2006). Progress in computational approach to drug development against sars. *Current Medicinal Chemistry*, 13(27), 3263–3270.

Chou, K.-C., Z.-C. Wu, and X. Xiao (2011). iloc-euk: a multi-label classifier for predicting the subcellular localization of singleplex and multiplex eukaryotic proteins. *PloS one*, 6(3).

Chowdhury, B. and G. Garai (2017). A review on multiple sequence alignment from the perspective of genetic algorithm. *Genomics*.

Contreras-Torres, E. (2018). Predicting structural classes of proteins by incorporating their global and local physicochemical and conformational properties into general chou's pseaac. *Journal of theoretical biology*, 454, 139–145.

Corder, G. W. and D. I. Foreman (2009). Nonparametric statistics: An introduction. *Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach*, 1–11.

Costantini, S. and A. M. Facchiano (2009). Prediction of the protein structural class by specific peptide frequencies. *Biochimie*, 91(2), 226–229.

Cuff, A. L., I. Sillitoe, T. Lewis, O. C. Redfern, R. Garratt, J. Thornton, and C. A. Orengo (2009). The cath classification revisited—architectures reviewed and new ways to characterize structural divergence in superfamilies. *Nucleic acids research*, 37(suppl 1), D310–D314.

Cutello, V., G. Nicosia, M. Pavone, and I. Prizzi (2011). Protein multiple sequence alignment by hybrid bio-inspired algorithms. *Nucleic acids research*, 39(6), 1980–1992.

Dai, Q., Y. Li, X. Liu, Y. Yao, Y. Cao, and P. He (2013). Comparison study on statistical features of predicted secondary structures for protein structural class prediction: From content to position. *BMC bioinformatics*, 14(1), 1.

Dayhoff, M., R. Schwartz, and B. Orcutt, 22 a model of evolutionary change in proteins. *In Atlas of protein sequence and structure*volume5. National Biomedical Research Foundation Silver Spring, MD, 1978, 345–352.

Dehzangi, A., S. P. Amnuaisuk, K. H. Ng, and E. Mohandesi, Protein fold prediction problem using ensemble of classifiers. *In International Conference on Neural Information Processing*. Springer, 2009.

Dehzangi, A., R. Heffernan, A. Sharma, J. Lyons, K. Paliwal, and A. Sattar (2015). Gram-positive and gram-negative protein subcellular localization by incorporating evolutionary-based descriptors into chou' s general pseaac. *Journal of theoretical biology*, 364, 284–294.

Dehzangi, A., K. Paliwal, J. Lyons, A. Sharma, and A. Sattar, Enhancing protein fold prediction accuracy using evolutionary and structural features. *In IAPR International Conference on Pattern Recognition in Bioinformatics*. Springer, 2013*a*.

Dehzangi, A., K. Paliwal, J. Lyons, A. Sharma, and A. Sattar (2014). A segmentation-based method to extract structural and evolutionary features for protein fold recognition. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 11(3), 510–519.

Dehzangi, A., K. Paliwal, A. Sharma, O. Dehzangi, and A. Sattar (2013*b*). A combination of feature extraction methods with an ensemble of different classifiers for protein structural class prediction problem. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(3), 564–575.

Dehzangi, A. and S. Phon-Amnuaisuk (2011). Fold prediction problem: The application of new physical and physicochemical-based features. *Protein and Peptide Letters*, 18(2), 174–185.

Dehzangi, A., S. Phon-Amnuaisuk, and O. Dehzangi (2010). Enhancing protein fold prediction accuracy by using ensemble of different classifiers. *Australian Journal of Intelligent Information Processing Systems*, 26(4), 32–40.

Deschavanne, P. and P. Tufféry (2009). Enhanced protein fold recognition using a structural alphabet. *Proteins: Structure, Function, and Bioinformatics*, 76(1), 129–137.

Devereux, J., P. Haeberli, and O. Smithies (1984). A comprehensive set of sequence analysis programs for the vax. *Nucleic acids research*, 12(1Part1), 387–395.

Ding, C. H. and I. Dubchak (2001). Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17(4), 349–358.

Ding, S., S. Yan, S. Qi, Y. Li, and Y. Yao (2014). A protein structural classes prediction method based on psi-blast profile. *Journal of Theoretical Biology*, 353, 19–23.

Dong, Q., S. Zhou, and J. Guan (2009). A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. *Bioinformatics*, 25(20), 2655–2662.

Du, P., S. Cao, and Y. Li (2009). Subchlo: predicting protein subchloroplast locations with pseudo-amino acid composition and the evidence-theoretic k-nearest neighbor (et-knn) algorithm. *Journal of theoretical biology*, 261(2), 330–335.

Edgar, R. (). http://www.drive5.com/bench/.

Edgar, R. C. (2004*a*). Muscle: a multiple sequence alignment method with reduced time and space complexity. *BMC bioinformatics*, 5(1), 113.

Edgar, R. C. (2004*b*). Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5), 1792–1797.

Efron, B. and G. Gong (1983). A leisurely look at the bootstrap, the jackknife, and cross-validation. *The American Statistician*, 37(1), 36–48.

Fan, G.-L. and Q.-Z. Li (2012). Predict mycobacterial proteins subcellular locations by incorporating pseudo-average chemical shift into the general form of chou's pseudo amino acid composition. *Journal of Theoretical Biology*, 304, 88–95.

Fei, X., Z. Dan, L. Lina, M. Xin, and Z. Chunlei (2018). Fpgasw: Accelerating large-scale smith–waterman sequence alignment application with backtracking on fpga linear systolic array. *Interdisciplinary Sciences: Computational Life Sciences*, 10(1), 176–188.

Feng, D.-F. and R. F. Doolittle (1987). Progressive sequence alignment as a prerequisitetto correct phylogenetic trees. *Journal of molecular evolution*, 25(4), 351–360.

Ferro, M., D. Salvi, S. Brugière, S. Miras, S. Kowalski, M. Louwagie, J. Garin, J. Joyard, and N. Rolland (2003). Proteomics of the chloroplast envelope membranes from arabidopsis thaliana. *Molecular & Cellular Proteomics*, 2(5), 325–345.

Fox, N. K., S. E. Brenner, and J.-M. Chandonia (2013). Scope: Structural classification of proteins—extended, integrating scop and astral data and classification of new structures. *Nucleic acids research*, 42(D1), D304–D309.

Fyshe, A., Y. Liu, D. Szafron, R. Greiner, and P. Lu (2008). Improving subcellular localization prediction using text classification and the gene ontology. *Bioinformatics*, 24(21), 2512–2517.

Ghanty, P. and N. R. Pal (2009). Prediction of protein folds: extraction of new features, dimensionality reduction, and fusion of heterogeneous classifiers. *IEEE transactions on nanobioscience*, 8(1), 100–110.

Gondro, C. and B. P. Kinghorn (2007). A simple genetic algorithm for multiple sequence alignment. *Genetics and Molecular Research*, 6(4), 964–982.

Gonnet, G. H., M. A. Cohen, and S. A. Benner (1992). Exhaustive matching of the entire protein sequence database. *Science*, 256(5062), 1443–1445.

Goodfellow, I., Y. Bengio, and A. Courville (2016). Deep learning— the mit press. *Cambridge, Massachusetts.*

Gromiha, M. M. (2005). A statistical model for predicting protein folding rates from amino acid sequence with structural class information. *Journal of chemical information and modeling*, 45(2), 494–501.

Hanke, G. T., S. Okutani, Y. Satomi, T. Takao, A. Suzuki, and T. Hase (2005). Multiple iso-proteins of fnr in arabidopsis: evidence for different contributions to chloroplast function and nitrogen assimilation. *Plant, Cell & Environment*, 28(9), 1146–1157.

Haralick, R. M., K. Shanmugam, and I. H. Dinstein (1973). Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6), 610–621.

Henikoff, S. and J. G. Henikoff (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22), 10915–10919.

Hu, J. and X. Yan (2012). Bs-knn: An effective algorithm for predicting protein subchloroplast localization. *Evolutionary Bioinformatics*, 8, EBO–S8681.

Ibrahim, W. and M. S. Abadeh (2017). Extracting features from protein sequences to improve deep extreme learning machine for protein fold recognition. *Journal of theoretical biology*, 421, 1–15.

Ibrahim, W. and M. S. Abadeh (2018). Protein fold recognition using deep kernelized extreme learning machine and linear discriminant analysis. *Neural Computing and Applications*, 1–14.

Jones, D. T. (1999). Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology*, 292(2), 195–202.

Joulin, A., E. Grave, P. Bojanowski, and T. Mikolov (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Karlik, B. and A. V. Olgac (2011). Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1(4), 111–122.

Katoh, K. and D. M. Standley (2013). Mafft multiple sequence alignment software version 7: improvements in performance and usability. *Molecular biology and evolution*, 30(4), 772–780.

Kavianpour, H. and M. Vasighi (2017). Structural classification of proteins using texture descriptors extracted from the cellular automata image. *Amino acids*, 49(2), 261–271.

Kavousi, K., B. Moshiri, M. Sadeghi, B. N. Araabi, and A. A. Moosavi-Movahedi (2011). A protein fold classifier formed by fusing different modes of pseudo amino acid composition via pssm. *Computational biology and chemistry*, 35(1), 1–9.

Kavousi, K., M. Sadeghi, B. Moshiri, B. N. Araabi, and A. A. Moosavi-Movahedi (2012). Evidence theoretic protein fold classification based on the concept of hyperfold. *Mathematical biosciences*, 240(2), 148–160.

Kaya, M., A. Sarhan, and R. Alhajj (2014). Multiple sequence alignment with affine gap by using multi-objective genetic algorithm. *Computer methods and programs in biomedicine*, 114(1), 38–49.

Kedarisetti, K. D., L. Kurgan, and S. Dick (2006). Classifier ensembles for protein structural class prediction with varying homology. *Biochemical and Biophysical Research Communications*, 348(3), 981–988.

Khan, M. I., M. S. Kamal, and L. Chowdhury (2016). Msupda: a memory efficient algorithm for sequence alignment. *Interdisciplinary Sciences: Computational Life Sciences*, 8(1), 84–94.

Kim, E. and J. Kececioglu (2008). Learning scoring schemes for sequence alignment from partial examples. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 5(4), 546–556.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kirk, P. R. and R. M. Leech (1972). Amino acid biosynthesis by isolated chloroplasts during photosynthesis. *Plant physiology*, 50(2), 228–234.

Klein, P. and C. Delisi (1986). Prediction of protein structural class from the amino acid sequence. *Biopolymers*, 25(9), 1659–1672.

Kong, L., L. Zhang, and J. Lv (2014). Accurate prediction of protein structural classes by incorporating predicted secondary structure information into the general form of chou's pseudo amino acid composition. *Journal of Theoretical Biology*, 344, 12–18.

Kumar, P., S. Bankapur, and N. Patil (2020). An enhanced protein secondary structure prediction using deep learning framework on hybrid profile based features. *Applied Soft Computing*, 86, 105926.

Kurgan, L. and K. Chen (2007). Prediction of protein structural class for the twilight zone sequences. *Biochemical and Biophysical Research Communications*, 357(2), 453–460.

Kurgan, L., K. Cios, and K. Chen (2008). Scpred: accurate prediction of protein structural class for sequences of twilight-zone similarity with predicting sequences. *BMC bioinformatics*, 9(1), 226.

Kurgan, L. and L. Homaeian, Prediction of secondary protein structure content from primary sequence alone–a feature selection based approach. *In International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2005.

Kurgan, L. A. and L. Homaeian (2006). Prediction of structural classes for protein sequences and domains - impact of prediction algorithms, sequence representation and homology, and test procedures on accuracy. *Pattern Recognition*, 39(12), 2323–2343.

Kuznetsov, I. B., Z. Gou, R. Li, and S. Hwang (2006). Using evolutionary and structural information to predict dna-binding sites on dna-binding proteins. *PROTEINS: Structure, Function, and Bioinformatics*, 64(1), 19–27.

Lassmann, T., O. Frings, and E. L. Sonnhammer (2008). Kalign2: high-performance multiple alignment of protein and nucleotide sequences allowing external features. *Nucleic acids research*, 37(3), 858–865.

Levitt, M. and C. Chothia (1976). Structural patterns in globular proteins. *Nature*, 261(5561), 552–558.

Li, Z., X. Zhou, Y. Lin, and X. Zou (2008). Prediction of protein structure class by coupling improved genetic algorithm and support vector machine. *Amino acids*, 35(3), 581–590.

Li, Z.-C., X.-B. Zhou, Z. Dai, and X.-Y. Zou (2009). Prediction of protein structural classes by chou's pseudo amino acid composition: approached using continuous wavelet transform and principal component analysis. *Amino acids*, 37(2), 415.

Liang, Y., S. Liu, and S. Zhang (2015). Prediction of protein structural class based on different autocorrelation descriptors of position-specific scoring matrix. *MATCH: Communications in Mathematical and in Computer Chemistry*, 73(3), 765–784.

Lin, K.-L., C.-Y. Lin, C.-D. Huang, H.-M. Chang, C.-Y. Yang, C.-T. Lin, C. Y. Tang, and D. F. Hsu (2007). Feature selection and combination criteria for improving accuracy in protein structure prediction. *IEEE Transactions on Nanobioscience*, 6(2), 186–196.

Lin, W.-Z., J.-A. Fang, X. Xiao, and K.-C. Chou (2013). iloc-animal: a multi-label learning classifier for predicting subcellular localization of animal proteins. *Molecular BioSystems*, 9(4), 634–644.

Liu, T., X. Geng, X. Zheng, R. Li, and J. Wang (2012). Accurate prediction of protein structural class using auto covariance transformation of psi-blast profiles. *Amino acids*, 42(6), 2243–2249.

Liu, T. and C. Jia (2010). A high-accuracy protein structural class prediction algorithm using predicted secondary structural information. *Journal of theoretical biology*, 267(3), 272–275.

Liu, T., X. Zheng, and J. Wang (2010*a*). Prediction of protein structural class for low-similarity sequences using support vector machine and psi-blast profile. *Biochimie*, 92(10), 1330–1334.

Liu, T., X. Zheng, and J. Wang (2010*b*). Prediction of protein structural class using a complexity-based distance measure. *Amino acids*, 38(3), 721–728.

Lodish, H., A. Berk, S. L. Zipursky, P. Matsudaira, D. Baltimore, and J. Darnell, *Molecular Cell Biology, 4th edition*. WH Freeman, 2000*a*.

Lodish, H., A. Berk, S. L. Zipursky, P. Matsudaira, D. Baltimore, and J. Darnell, *The three roles of RNA in protein synthesis, Molecular Cell Biology, 4th edition*. WH Freeman, 2000*b*. URL `https://www.ncbi.nlm.nih.gov/books/NBK21603/`.

Lyons, J., N. Biswas, A. Sharma, A. Dehzangi, and K. K. Paliwal (2014). Protein fold recognition by alignment of amino acid residues using kernelized dynamic time warping. *Journal of theoretical biology*, 354, 137–145.

Lyons, J., A. Dehzangi, R. Heffernan, Y. Yang, Y. Zhou, A. Sharma, and K. Paliwal (2015). Advancing the accuracy of protein fold recognition by utilizing profiles from hidden markov models. *IEEE transactions on nanobioscience*, 14(7), 761–772.

Lyons, J., K. K. Paliwal, A. Dehzangi, R. Heffernan, T. Tsunoda, and A. Sharma (2016). Protein fold recognition using hmm–hmm alignment and dynamic programming. *Journal of theoretical biology*, 393, 67–74.

M.A. Larkin, N. B. R. C. P. M. H. M. F. V. I. W. A. W. R. L. J. T. T. G. D. H., G. Blackshields (2007). Clustal w and clustal x version 2.0. *bioinformatics*, 23(21), 2947–2948.

Mao, Z., G.-S. Han, and T.-T. Wang, Effects of amino acid classification on prediction of protein structural classes. *In Fuzzy Systems and Knowledge Discovery (FSKD), 2013 10th International Conference on*. IEEE, 2013.

Maton, A., D. Lahart, J. Hopkins, M. Q. Warner, S. Johnson, and J. D. Wright, *Cells: Building blocks of life*. Pearson Prentice Hall, 1997.

McGuffin, L. J., K. Bryson, and D. T. Jones (2000). The psipred protein structure prediction server. *Bioinformatics*, 16(4), 404–405.

Melkikh, A. V., V. D. Seleznev, and O. I. Chesnokova (2010). Analytical model of ion transport and conversion of light energy in chloroplasts. *Journal of theoretical biology*, 264(3), 702–710.

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mizianty, M. J. and L. Kurgan (2009). Modular prediction of protein structural classes from sequences of twilight-zone identity with predicting sequences. *BMC bioinformatics*, 10(1), 414.

Moore, F. and D. Shephard (1978). Chloroplast autonomy in pigment synthesis. *Protoplasma*, 94(1-2), 1–17.

Murzin, A. G., S. E. Brenner, T. Hubbard, and C. Chothia (1995). Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4), 536–540.

Naznin, F., R. Sarker, and D. Essam (2012). Progressive alignment method using genetic algorithm for multiple sequence alignment. *IEEE Transactions on Evolutionary Computation*, 16(5), 615–631.

Needleman, S. B. and C. D. Wunsch (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3), 443–453.

Nguyen, K. D. and Y. Pan (2011). An improved scoring method for protein residue conservation and multiple sequence alignment. *IEEE transactions on nanobioscience*, 10(4), 275–285.

Ningbo, L. and H. Hua (2017). An artificial neural network classifier for the prediction of protein structural classes. *Inl Jnl of Curr Engg and Tech*, 7(3).

Notredame, C. (2002). Recent progress in multiple sequence alignment: a survey. *Pharmacogenomics*, 3(1), 131–144.

Notredame, C. and D. G. Higgins (1996). Saga: sequence alignment by genetic algorithm. *Nucleic acids research*, 24(8), 1515–1524.

Notredame, C., D. G. Higgins, and J. Heringa (2000). T-coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, 302(1), 205–217.

Ortuno, F., J. P. Florido, J. M. Urquiza, H. Pomares, A. Prieto, and I. Rojas, Optimization of multiple sequence alignment methodologies using a multiobjective evolutionary algorithm based on nsga-ii. *In IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2012.

Oyola, S. O., T. D. Otto, Y. Gu, G. Maslen, M. Manske, S. Campino, D. J. Turner, B. MacInnis, D. P. Kwiatkowski, H. P. Swerdlow, and M. A. Quail (2012). Optimizing illumina next-generation sequencing library preparation for extremely at-biased genomes. *BMC genomics*, 13(1), 1.

Paliwal, K. K., A. Sharma, J. Lyons, and A. Dehzangi (2014*a*). Improving protein fold recognition using the amalgamation of evolutionary-based and structural based information. *BMC bioinformatics*, 15(16), S12.

Paliwal, K. K., A. Sharma, J. Lyons, and A. Dehzangi (2014*b*). A tri-gram based feature extraction technique using linear probabilities of position specific scoring matrix for protein fold recognition. *IEEE transactions on nanobioscience*, 13(1), 44–50.

Pauling, L., R. B. Corey, and H. R. Branson (1951). The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proceedings of the National Academy of Sciences*, 37(4), 205–211.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Pennington, J., R. Socher, and C. Manning, Glove: Global vectors for word representation. *In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.

Post-Beittenmiller, D., G. Roughan, and J. B. Ohlrogge (1992). Regulation of plant fatty acid biosynthesis: Analysis of acyl-coenzyme a and acyl-acyl carrier protein substrate pools in spinach and pea chloroplasts. *Plant Physiology*, 100(2), 923–930.

Provencher, S. W. and J. Gloeckner (1981). Estimation of globular protein secondary structure from circular dichroism. *Biochemistry*, 20(1), 33–37.

Qin, Y., X. Zheng, J. Wang, M. Chen, and C. Zhou (2015). Prediction of protein structural class based on linear predictive coding of psi-blast profiles. *Open Life Sciences*, 10(1).

Rahal, I. and J. Walz (2018). Secondary protein structure prediction combining protein structural class, relative surface accessibility, and contact number. *International Journal of Data Science*, 3(1), 68–85.

Raicar, G., H. Saini, A. Dehzangi, S. Lal, and A. Sharma (2016). Improving protein fold recognition and structural class prediction accuracies using physicochemical properties of amino acids. *Journal of theoretical biology*, 402, 117–128.

Remmert, M., A. Biegert, A. Hauser, and J. Söding (2012). Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nature methods*, 9(2), 173.

Rubio-Largo, Á., M. A. Vega-Rodríguez, and D. L. González-Álvarez (2016). A hybrid multiobjective memetic metaheuristic for multiple sequence alignment. *IEEE Transactions on Evolutionary Computation*, 20(4), 499–514.

Sahu, S. S. and G. Panda (2010). A novel feature representation method based on chou's pseudo amino acid composition for protein structural class prediction. *Computational biology and chemistry*, 34(5), 320–327.

Saini, H., G. Raicar, A. Sharma, S. Lal, A. Dehzangi, J. Lyons, K. K. Paliwal, S. Imoto, and S. Miyano (2015). Probabilistic expression of spatially varied amino acid dimers into general form of chou' s pseudo amino acid composition for protein fold recognition. *Journal of theoretical biology*, 380, 291–298.

Saitou, N. and M. Nei (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, 4(4), 406–425.

Sedgwick, P. (2012). Pearson's correlation coefficient. *Bmj*, 345(7).

Segrest, J. P., M. K. Jones, A. E. Klon, C. J. Sheldahl, M. Hellinger, H. De Loof, and S. C. Harvey (1999). A detailed molecular belt model for apolipoprotein ai in discoidal high density lipoprotein. *Journal of Biological Chemistry*, 274(45), 31755–31758.

Sharma, A., J. Lyons, A. Dehzangi, and K. K. Paliwal (2013*a*). A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition. *Journal of theoretical biology*, 320, 41–46.

Sharma, A., K. K. Paliwal, A. Dehzangi, J. Lyons, S. Imoto, and S. Miyano (2013*b*). A strategy to select suitable physicochemical attributes of amino acids for protein fold recognition. *BMC bioinformatics*, 14(1), 233.

Shen, H.-B. and K.-C. Chou (2006). Ensemble classifier for protein fold pattern recognition. *Bioinformatics*, 22(14), 1717–1722.

Shen, H.-B. and K.-C. Chou (2009). Predicting protein fold pattern with functional domain and sequential evolution information. *Journal of Theoretical Biology*, 256(3), 441–446.

Shen, H.-B., J.-N. Song, and K.-C. Chou (2009). Prediction of protein folding rates from primary sequence by fusing multiple sequential features. *Journal of Biomedical Science and Engineering (JBiSE)*, 2, 136–143.

Shi, S.-P., J.-D. Qiu, X.-Y. Sun, J.-H. Huang, S.-Y. Huang, S.-B. Suo, R.-P. Liang, and L. Zhang (2011). Identify submitochondria and subchloroplast locations with pseudo amino acid composition: approach from the strategy of discrete wavelet transform feature extraction. *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research*, 1813(3), 424–430.

Shyu, C., L. Sheneman, and J. A. Foster (2004). Multiple sequence alignment with evolutionary computation. *Genetic Programming and Evolvable Machines*, 5(2), 121–144.

Sievers, F., D. Dineen, A. Wilm, and D. G. Higgins (2013). Making automated multiple alignments of very large numbers of protein sequences. *Bioinformatics*, 29(8), 989–995.

Sievers, F., A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Söding, J. D. Thompson, and D. G. Higgins (2011). Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. *Molecular systems biology*, 7(1), 539.

Smith, R. F. and T. F. Smmith (1992). Pattern-induced multi-sequence alignment (pima) algorithm employing secondary structure-dependent gap penalties for use in comparative protein modelling. *Protein Engineering, Design and Selection*, 5(1), 35–41.

Smith, T. F. and M. S. Waterman (1981). Identification of common molecular subsequences. *Journal of molecular biology*, 147(1), 195–197.

Sneath, P. and R. Sokal (1973). Numerical taxonomy freeman san francisco.

Taguchi, Y. and M. M. Gromiha (2007). Application of amino acid occurrence for discriminating different folding types of globular proteins. *BMC bioinformatics*, 8(1), 404.

Taheri, J. and A. Y. Zomaya (2009). Rbt-ga: a novel metaheuristic for solving the multiple sequence alignment problem. *Bmc Genomics*, 10(1), S10.

Taylor, W. R. (1988). A flexible method to align large numbers of biological sequences. *Journal of Molecular Evolution*, 28(1-2), 161–169.

Thompson, J. D., T. J. Gibson, F. Plewniak, F. Jeanmougin, and D. G. Higgins (1997). The clustal_x windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucleic acids research*, 25(24), 4876–4882.

Thompson, J. D., D. G. Higgins, and T. J. Gibson (1994). Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, 22(22), 4673–4680.

Thompson, J. D., F. Plewniak, and O. Poch (1999). A comprehensive comparison of multiple sequence alignment programs. *Nucleic acids research*, 27(13), 2682–2690.

Tung, C.-W., C. Liaw, S.-J. Ho, and S.-Y. Ho, Prediction of protein subchloroplast locations using random forests. *In Proceeding of World Academy of Science, Engineering and Technology*volume4. Citeseer, 2010.

UniProtKB/Swiss-Prot (2019). Uniprotkb/swiss-prot protein knowledgebase release 2019-07 statistics. URL `https://web.expasy.org/docs/relnotes/relstat.html`.

Valdar, W. S. J. (2001). *Residue conservation in the prediction of protein-protein interfaces*. Ph.D. thesis, University College London (University of London).

Van Walle, I., I. Lasters, and L. Wyns (2004*a*). Align-m—a new algorithm for multiple alignment of highly divergent sequences. *Bioinformatics*, 20(9), 1428–1435.

Van Walle, I., I. Lasters, and L. Wyns (2004*b*). Sabmark—a benchmark for sequence alignment that covers the entire known fold space. *Bioinformatics*, 21(7), 1267–1268.

Wan, S., M.-W. Mak, and S.-Y. Kung (2016*a*). Sparse regressions for predicting and interpreting subcellular localization of multi-label proteins. *BMC bioinformatics*, 17(1), 97.

Wan, S., M.-W. Mak, and S.-Y. Kung (2016*b*). Transductive learning for multi-label protein subchloroplast localization prediction. *IEEE/ACM transactions on computational biology and bioinformatics*, 14(1), 212–224.

Wang, L. and T. Jiang (1994). On the complexity of multiple sequence alignment. *Journal of computational biology*, 1(4), 337–348.

Wang, X., W. Zhang, Q. Zhang, and G.-Z. Li (2015). Multip-schlo: multi-label protein subchloroplast localization prediction with chou's pseudo amino acid composition and a novel multi-label classifier. *Bioinformatics*, 31(16), 2639–2645.

Wang, Z. and C. Benning (2012). Chloroplast lipid synthesis and lipid trafficking through er–plastid membrane contact sites. *Biochem. Soc. Trans.*, 40(2), 457–463.

Wang, Z.-X. and Z. Yuan (2000). How good is prediction of protein structural class by the component-coupled method? *Proteins: Structure, Function, and Bioinformatics*, 38(2), 165–175.

Waterman, M. S., T. F. Smith, and W. A. Beyer (1976). Some biological sequence metrics. *Advances in Mathematics*, 20(3), 367–387.

Xia, X.-Y., M. Ge, Z.-X. Wang, and X.-M. Pan (2012). Accurate prediction of protein structural class. *PLoS One*, 7(6), e37653.

Xiao, X., X. Cheng, S. Su, Q. Mao, and K.-C. Chou (2017). ploc-mgpos: incorporate key gene ontology information into general pseaac for predicting subcellular localization of gram-positive bacterial proteins. *Natural Science*, 9(09), 330.

Yan, K., Y. Xu, X. Fang, C. Zheng, and B. Liu (2017). Protein fold recognition based on sparse representation based classification. *Artificial intelligence in medicine*, 79, 1–8.

Yang, J.-Y. and X. Chen (2011). Improving taxonomy-based protein fold recognition by using global and local features. *Proteins: Structure, Function, and Bioinformatics*, 79(7), 2053–2064.

Yang, J.-Y., Z.-L. Peng, Z.-G. Yu, R.-J. Zhang, V. Anh, and D. Wang (2009). Prediction of protein structural classes by recurrence quantification analysis based on chaos game representation. *Journal of Theoretical Biology*, 257(4), 618–626.

Yang, T., V. Kecman, L. Cao, C. Zhang, and J. Z. Huang (2011). Margin-based ensemble classifier for protein fold recognition. *Expert Systems with Applications*, 38(10), 12348–12355.

Ying, Y., K. Huang, and C. Campbell (2009). Enhanced protein fold recognition through a novel data integration approach. *BMC bioinformatics*, 10(1), 267.

Yu, D.-J., J. Hu, X.-W. Wu, H.-B. Shen, J. Chen, Z.-M. Tang, J. Yang, and J.-Y. Yang (2013). Learning protein multi-view features in complex space. *Amino acids*, 44(5), 1365–1379.

Yuan, M., Z. Yang, G. Huang, and G. Ji (2018). A novel feature selection method to predict protein structural class. *Computational biology and chemistry*.

Zaman, R., S. Y. Chowdhury, M. A. Rashid, A. Sharma, A. Dehzangi, and S. Shatabda (2017). Hmmbinder: Dna-binding protein prediction using hmm profile based features. *BioMed research international*, 2017.

Zhang, L., X. Zhao, and L. Kong (2014). Predict protein structural class for low-similarity sequences by evolutionary difference information into the general form of chou's pseudo amino acid composition. *Journal of theoretical biology*, 355, 105–110.

Zhang, M.-L. and L. Wu (2014). Lift: Multi-label learning with label-specific features. *IEEE transactions on pattern analysis and machine intelligence*, 37(1), 107–120.

Zhang, S. and X. Duan (2018). Prediction of protein subcellular localization with oversampling approach and chou's general pseaac. *Journal of theoretical biology*, 437, 239–250.

Zhang, S., F. Ye, and X. Yuan (2012). Using principal component analysis and support vector machine to predict protein structural class for low-similarity sequences via pssm. *Journal of Biomolecular Structure and Dynamics*, 29(6), 1138–1146.

Zhang, T.-L., Y.-S. Ding, and K.-C. Chou (2008). Prediction protein structural classes with pseudo-amino acid composition: approximate entropy and hydrophobicity pattern. *Journal of theoretical biology*, 250(1), 186–193.

Zheng, X., C. Li, and J. Wang (2010). An information-theoretic approach to the prediction of protein structural class. *Journal of computational chemistry*, 31(6), 1201–1206.

Zhou, G.-P. (1998). An intriguing controversy over protein structural class prediction. *Journal of protein chemistry*, 17(8), 729–738.

Zhou, G.-P. and K. Doctor (2003). Subcellular location prediction of apoptosis proteins. *Proteins: Structure, Function, and Bioinformatics*, 50(1), 44–48.

Zhu, H., Z. He, and Y. Jia (2016). A novel approach to multiple sequence alignment using multiobjective evolutionary algorithm based on decomposition. *IEEE journal of biomedical and health informatics*, 20(2), 717–727.

Zhu, X.-J., C.-Q. Feng, H.-Y. Lai, W. Chen, and L. Hao (2019). Predicting protein structural classes for low-similarity sequences by evaluating different features. *Knowledge-Based Systems*, 163, 787–793.

Publications

Journal Papers

1. **Sanjay Bankapur** and Nagamma Patil, "ProgSIO-MSA: Progressive based Single Iterative Optimization Framework for Multiple Sequence Alignment using an Effective Scoring System". Journal of Bioinformatics and Computational Biology, Vol. 18, p.2050005, World Scientific, 2020. [SCIE and SCOPUS indexed] (**Published**) Doi: `https://doi.org/10.1142/S0219720020500055`

2. **Sanjay Bankapur** and Nagamma Patil, "Enhanced Protein Structural Class Prediction using Effective Feature Modeling and Ensemble of Classifiers". IEEE/ACM Transactions on Computational Biology and Bioinformatics, (In Press, 2020). [SCIE and SCOPUS indexed] (**Published**)
Doi: `https://doi.org/10.1109/TCBB.2020.2979430`

3. Prince Kumar, **Sanjay Bankapur** and Nagamma Patil, "An enhanced protein secondary structure prediction using deep learning framework on hybrid profile based features". Applied Soft Computing, Vol. 86, p.105926, Elsevier, 2020. [SCIE and SCOPUS indexed] (**Published**)
Doi: `https://doi.org/10.1016/j.asoc.2019.105926`

4. **Sanjay Bankapur** and Nagamma Patil, "An Enhanced Protein Fold Recognition for Low Similarity Datasets using Convolutional and Skip-Gram Features with Deep Neural Network". IEEE Transactions on NanoBioscience, vol. 20, no. 1, pp. 42-49, Jan. 2021. [SCIE and SCOPUS indexed] (**Published**)
Doi: `https://doi.org/10.1109/TNB.2020.3022456`

5. **Sanjay Bankapur** and Nagamma Patil, "An Effective Multi-Label Protein Sub-Chloroplast Localization Prediction by Skipped-grams of Evolutionary Profiles using Deep Neural Network". IEEE/ACM Transactions on Computational Biology and Bioinformatics, (In Press, 2020). [SCIE and SCOPUS indexed] (**Published**)
Doi: `https://doi.org/10.1109/TCBB.2020.3037465`

6. Abhilash Venkatesh, Shrinivas V. Shanbhag, **Sanjay Bankapur**, and Nagamma Patil, "Multi-Label Protein Sub-Chloroplast Localisation Prediction using Binary Relevance Framework and Machine Learning Techniques". International Journal of Data Mining and Bioinformatics, Inderscience. [SCIE and SCOPUS indexed] (**Communicated**)

Conference Papers

1. **Sanjay Bankapur** and Nagamma Patil, "Efficient and Effective Multiple Protein Sequence Alignment Model Using Dynamic Progressive Approach with Novel Look Back Ahead Scoring System". In Proceedings of the 7th International Conference on Pattern Recognition and Machine Intelligence 2017 (PReMI '17). ISI Kolkata, Dec 5-8 2017 (pp. 397-404). Springer. [SCOPUS indexed]
Doi: `https://doi.org/10.1007/978-3-319-69900-4_50`

2. **Sanjay Bankapur** and Nagamma Patil, "Position-Residue Specific Dynamic Gap Penalty Scoring Strategy for Multiple Sequence Alignment". In Proceedings of the 8th International Conference on Computational Systems-Biology and Bioinformatics (CSBio '17). Nha Trang, Vietnam, Dec 7-8 2017 (pp. 42-45). ACM. [SCOPUS indexed]
   Doi: `https://doi.org/10.1145/3156346.3156354`

3. **Sanjay Bankapur** and Nagamma Patil, "Protein Secondary Structural Class Prediction Using Effective Feature Modeling and Machine Learning Techniques". In 2018 IEEE 18th International Conference on Bioinformatics and Bioengineering (BIBE). Taichung, Taiwan. Oct 29-31 2018 (pp. 18-21). IEEE. [SCOPUS indexed] (**CORE C**).
   Doi: `https://doi.org/10.1109/BIBE.2018.00012`

Curriculum Vitae

Mr. Sanjay S. Bankapur
Full-Time Research Scholar
Department of Information Technology
National Institute of Technology Karnataka
P.O. Srinivasanagar, Surathkal
Mangalore-575 025

Permanent Address

Sanjay S. Bankapur
#2160, M.C.C. A Block
4th Main Road, Davangere -577004
Davangere District, Karnataka, India.
Email: sanjaybankapur.mit@gmail.com
Mobile: +91-9972244997.

Academic Records

1. M.Tech in Computer Science and Engineering from IIIT-Hyderabad, India, 2011-2013.
2. B.E. in Computer Science and Engineering from B.I.E.T, Davangere, Karnataka, India, 2001-2005.

Professional Experience

1. Worked as a Software Engineer in Tech Mahindra Ltd., Pune from 2005-2007.
2. Worked as a Senior Software Engineer in Birlasoft Pvt. Ltd., Bangalore from 2007-2009.
3. Worked as a Team Lead in Accenture, Bangalore from 2009-2011.
4. Worked as a Senior Oracle Consultant in eVerge, Bangalore from 2014-2014.
5. Worked as an Assistant Professor in Manipal Institute of Technology, Manipal from 2014-2014.

Research Interests

Algorithms, Data Mining, Bioinformatics, Machine Learning, Soft Computing, Computational Biology.