Subcellular Localization of Multi-class Proteins Using Label Power-set Encoding

Hasnaeen Ferdous Bin Hashem
M.Sc in CSE
ID : 012 152 006
United International University

A thesis submitted for the degree of
Masters of Science In Computer Science and Engineering
May 2018

Declaration

I, Hasnaeen Ferdous Bin Hashem, declare that this thesis titled, Thesis Title and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a MSc/ BSc degree at United International University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at United International University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

____

Date:

____

Hasnaeen Ferdous Bin Hashem

Certificate

I do hereby declare that the research works embodied in this thesis entitled Thesis Title is the outcome of an original work carried out by Student Name under my supervision.

I further certify that the dissertation meets the requirements and the standard for the degree of MSc/ BSc in Computer Science and Engineering.

Signed:

Date:

SHATABDA, SWAKKHAR, Ph.D.
Associate Professor
Department of Computer Science and Engineering,
United International University,
Dhaka-1209, Bangladesh.

Publications

Work relating to the research presented in this thesis has been published/ submitted by the author in the following peer-reviewed journals and conferences:

Paper Title : ”SUBCELLULAR LOCALIZATION OF GRAM-NEGATIVE PROTEINS USING LABEL POWER-SET ENCODING”.

Author : Hasnaeen Ferdous Bin Hashem, Raihan Uddin, Swakkhar Shatabda.
Published : International Conference on Emerging Technologies in Data Mining and Information Security (IEMIS 2018), Kolkata, India.

The conference proceedings will be published in Springer Advances in Intelligent Systems and Computing (AISC) Series, now indexed by: ISI Proceedings, DBLP. Ulrich's, EI-Compendex, SCOPUS, Zentralblatt Math, MetaPress, Springerlink.

Acknowledgments

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

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

I also thank for my another author of published paper Raihan Uddin. He helped me a lot to initially understand the features and move forward with my own study and research.

My heartfelt gratitude and appreciation to the panel of Examiners Dr. Mohammad Nurul Huda, Professor, Dept of CSE, UIU, for his constructive comments suggestions and critiquing.

I would like to express my deepest thanks to Dr. Dewan Md. Farid, Associate Professor, Dept of CSE, UIU, and HASAN, RUBABA , Lecturer, Dept of CSE, UIU for their favorable responses regarding the study.

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

Abstract

As knowledge is essential in all research or even research initiative. Therefore, biologists always try to know where a protein resides in a cell . They can elucidate the functions of the protein with this revelation. Armed with marvelous accomplishment of upcoming and ongoing large-scale genome sequencing projects, an exponentially growing number of new protein sequences have been discovered. Rather than using expensive lab experiments, computational methods are far more effective to automatically and accurately identify the subcellular locations of these proteins.

This book proposes an efficient multi-label predictor method, namely label powerset encoding, for predicting the subcellular localization of multi-location proteins. Briefly, on two recently published gram negative bacteria and plant datasets.

Bacterial proteins play an important role in cell biology due to their importance in drug design and antibiotics research. The localization of bacterial proteins are very important since the function of a protein is closely linked with its location. A single gram negative bacteria proteins can be located in multiple locations in a protein. Prediction of subcellular locations of gram negative bacteria proteins is thus far more challenging and difficult.

In this book, we proposed a novel method for subcellular localization of gram negative bacteria and plant protein dataset. Our method uses label power-set encoding scheme for the associated multi-label classification problem. Using a set of effective features also used in the literature our encoding significantly improves over the traditional approaches on several base classifiers. Our method was tested using a standard benchmark dataset and showed promising results.

Contents

Chapter 1

Introduction

The first section of this book is dedicated to present an overview of the motivation of our research and the aims and objectives identfied during literature review. We also give a brief summary of the methodology applied to achieve the research aims and present a summary of the contributions made by our research reported in this thesis.

1.1 Protein Sub-Cellular Localization

Locations of proteins in a cell are closely related to their functions within a cell. Subcellular localization of proteins is very important for the knowledge of metabolic pathways and signaling biological processes within the cell. Bacterial proteins can be broadly categorized into two types: gram-positive and gram-negative. Another dataset have been analyzed for comparison purpose, That dataset is plat-protein dataset.

Protein sub-cellular localization prediction is a term refers to the work that involves predicting the whereabouts of any protein that resides in a cell. Generally, the available prediction tools receive information as input about a protein. For example, this input can be a protein sequence of amino acids. With this given input, the tools can produce a predicted location within the cell as output. The output prediction of location can be the parts of a cell such as the nucleus, Endoplasmic reticulum, Golgi bodies, extracellular space or other organelles. The sole purpose of this study is to develop a more efficient tool that can predict the location of protein in a cell. This prediction of proteins sub-cellular localization is a very important aspect in bioinformatics. Its importance spread in the topics of prediction of protein function and genome annotation. Also it is very relevant for its capacity to aid in identification of drug targets. Protein sub-cellular localization is widely considered as a significant step for protein function prediction and modern drug design.

The question can be asked that if the modern science discovered many of the cells protein location, why the prediction tools are needed? The answer is not only simple but also very relevant. Though it is relatively easier than before to map proteins location with modern tools and other mapping methods. But, this process is still very lengthy and costly. Also, the amount of cells in the wonderful and variant world is far too great to cover with this much resource and time. Hence, the prediction comes in. With more efficient predicator, more quickly we can predict the proteins sub-cellular localization. With this information we can also move quickly to decide what should be the best way to use the cell or know its characteristics.

Identifying proteins sub-cellular location and functions are one of the fundamental goals in cell biology [16]. Detailed insight regarding sub-cellular location may reveal useful information. These information may bear the characteristics of proteins functions. Bacteria proteins hold an important position in the field of cell biology. These Bacteria proteins are very special in several ways. They have a unique duel role in a cell. Biologically Bacterial proteins are both harmful and useful [88]. Among the numerous types of bacteria, some bacteria plays active role to cause various diseases. Contrastingly, some others act as catalyst in biological interactions for betterment of health and biological harmony. Human race found a better way to put Bacteria in use. For example, some bacteria are frequently used to create antibiotics. Bacteria is a prokaryotic micro-organism that can be divided into two groups. Those are gram-positive and gram-negative [71]. This category is made by a special test. While a gram-stain test is done on a bacteria, if it is Gram-positive, then the bacteria stained dark blue or violet. If it is gram-negative, then the bacteria cannot retain the stain. Instead it takes up the counter-stain and appear red or pink [88].

As clearly mentioned and discussed in various well-known reviews [14], within the recent decade or more so, many web-servers were designed and dedicatedly deployed for predicting the sub-cellular localization of proteins. This predictions are made for single site and multiple sites. The sites are predicted based on their sequence information. They can be roughly classified into two series [14].

The categories are PLoc series and iLoc series. The PLoc series contains six web-servers [71], [18], [72], [19], [73], [74] to deal with eukaryotic, human, plants, Gram positive micro-organisms, Gram negative micro-organisms and virus proteins. The iLoc series contains another seven web-servers [90], [21], [87] , [54], [88], [90], [89] to deal with eukaryotic, human, plant, animal, Gram positive, Gram negative, and virus proteins, respectively. It is very unique and interesting that most proteins is known to be able to function only in specific place in a biological cell (e.g golgi bodies, ribosome). Contrastingly, there are some evidence found that some of the proteins are able to function in various places within a cell. For any given protein or protein type to work and function properly, the protein needs to be in the specific locations into a cell. Otherwise, the protein will malfunction in all other places. Therefore, the recently synthesized proteins have a critically important role where proteins are placed in correct sub-cellular compartments [96].

The sub-cellular location of a protein also can be detected and determined by various biological experiments. But it is as said earlier, those methods are very costly and exceedingly time consuming. The newly discovered sequence of proteins are increasing exponentially. The sheer number of sequenced proteins that are discovered every month demands faster process.The slower rate of determining protein structure using experimental approaches indicates a crucial demand for a fast-computational approach. Where this approach will be bale to shorten the time and also it will be an acceptable alternative to experimental methods.

Recently, the ever-growing popular computational methods are becoming increasingly important and recognized in the relevant fields. Researches worldwide clearly prefer using predication system to find out the sub-cellular localization of proteins [83], [84], [80], [81]. Faster computational approaches discourse the challenges of expensive and time-consuming lab-based sophisticated experimental methods. A very wide range of pattern recognition tactics has been practiced to unravel sub-cellular localization delinquencies. These pattern recognition tactics have given the most hopeful outcome.

To mention or point our briefly, the said approaches either contains classifier development or feature extraction development. The performance of pattern recognition technique to discourse protein sub-cellular localization prediction task rests on the classification technique and features that are being used. Up until now, several good classifiers have been developed and thoroughly analyzed. These are as follows: Artificial Neural Network (ANN), K-Nearest Neighbor (KNN) [31], Bayesian classifiers, Linear Discriminant Analysis (LDA), Hidden Markov Model (HMM), Nave Bayes [29], Support Vector Machine (SVM) [30], [37], and ensemble of classifiers.

Amongst these classifiers Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) based classifiers have given the most promising results [37]. Various reputed studies have revealed that most significant improvement in a predication system is accomplished by developing feature extraction method rather than refining the classifiers.

1.2 Motivation

Gram positive bacteria take purple color during gram stain test. On the contrary due to the thinner peptidoglycan layer of gram negative bacteria, they take up the counter stain and appear red or pink. In vitro localization methods like fluorescent microscopy [1] are very time consuming and expensive. This is why the computational approaches are becoming very popular to predict subcellular localization of bacterial proteins. Proteins are located in various locations within a cell. Supervised learning methods used for protein subcellular localization defines the problem as a multi-class classification problem. Many supervised learning methods have been proposed in the literature to handle protein subcellular localization problem. Most successful methods were: Support Vector Machines (SVM) [2], Artificial Neural Networks (ANN) [3], Naive Basian Classifiers [4],Decision Tree [5] and ensemble of classifiers [6]. An additional difficulty to this problem is added by the fact that a single protein can be located at multiple locations which makes the problem a multi-label classification problem [7]. Formerly practiced or used features were mostly sequence based or PSSM profile based as far as we can see the various studies. In this area of research the researchers refrain from using SPIDER information which gives structural information. In literature [59], we can see that the structure based features were used in some cases. Judging by the works and impacts of the study, we believe structure based features play an important role in protein sub-cellular localization as structural based feature comprises a vast amount of information about a protein.

There are different kinds of features as well as several features together used in different literatures. But for comprehensive analysis of these features were not found among the notable works of this arena. Again, most of the literatures tend to not use previous literature features or did not combine new features with older ones. It is unclear whether the practice is fruitful or deliberate bias. However, we believe combining some new features with some of the well-known old ones may increase the desired performance of the prediction. We are also hopeful that if we select best features found in previous studies, it will be able to provide a remarkable result. More elaborately, combining all the effective features used previously with all our newly extracted features from evolutionary based information or PSSM and structural based information or SPIDER, that will significantly increase the sub-cellular localization prediction rate.

1.3 Research Goals

Traditionally, the researchers tend to employ binary relevance encoding for gripping the multi-label classification problem of protein subcellular localization. One of the key disadvantages of the said method is that they have learned multiple classifiers either for each label or for each nth location [8, 2, 9].

These expressed methods definitely increases the time complexity of the training phase of our work. In this research, we suggest a label powerset based encoding for protein subcellular localization of gram negative bacteria proteins. Using this encoding scheme, our method learns only a single model. If we compare our method from the others, it is distinguished that all other studies have used multiple models together for the same thing. To the best of our knowledge, this is the first application of label powerset encoding for prediction of gram negative protein sub-cellular localization problem.

We have rigorously tested the full effectiveness of our method. We tested it by using different classifiers with the use of a standard set of features. On a standard benchmark dataset, our method was able to significantly improve over the state-of-the-art prediction methods for gram negative protein subcellular localization.

1.4 Methodology

The scope of the research is already explained clearly is the previous sections. Also the objectives of this study are discussed at length for better understanding our approach to achieve the result we want. For clarity, once again we want to express our goal, our goal is to develop effective strategies that improves the performance of predicting protein sub-cellular localization for plant and gram-negative bacteria protein dataset using label powerset encoding. Throughout this research, we were fully focused on improving label powerset encoding result. In this sub-cellular prediction problem, we have availed plant and gram-negative bacteria protein dataset to work on.

1.5 Research Contribution

The key contributions of the research are as follows:

1. Choose the multi class protein dataset.
2. As multi class dataset we used plant and gram negative dataset.
3. We have extracted new features from PSSM and Combined all the 7 features to analysis

4. We have done a comprehensive analysis with different classifier on machine learning context
5. Comparing binary relevance with label power-set encoding.

1.6 Thesis Organization

This book has been organized according to the structure of a good bed time story! We wanted to engage audience with our work through the relevance of the work. Therefore, the thesis book has been divided into five (5) major portions. Those are Introduction, Background, Materials and Methods, Result Discussion (Experimentation, Result and Discussion) and Conclusion.

In the introduction part we covered basics of this study. We discussed about Protein Sub-Cellular Localization and our motivation to choose this topic. We briefly discussed our Research Goals and outlined the Methodology around it.

Next we move onto 'Background' of this study. This section thoroughly examined the main curiosity regarding this paper's topic, 'Protein' and tools that helps determine Protein Sub-cellular Localization. During this study we have visited Protein's importance in biology and why we are trying to have more firm grasp on its Sub-cellular Localization. Then we explained Position Specific Score Matrices (PSSM), Machine Learning Background and Support Vector Machine (SVM). Then, we presented the details on Decision Tree and Random Forest. In the last portion of this section, we provided an elaborate Literature Review for general discussion regarding similar works in the past.

Within the Materials and Method part, we explained the Data Set. Also, we discussed Normalization of PSSM and Constructing the Consensus Sequence. Then, we moved to present various Feature Extraction Methods such as Composition Feature, PSSM-SD Feature, PSSM-SAC Feature, Auto Co-variance Feature, One-Lead Bi-Gram Feature, Torsional Angles Composition and Auto Co-variances of Probabilities. In the last section of this part, we discussed Label Transformation Method which is the main tool of this study.

The result section has three parts with Experimentation, Result and Discussion. First we explained how we extracted features in Feature Extraction. Then we discussed our Choosing Classifier and Parameter Tuning. We moved to explain Choosing Validation Method and Sensitivity, Speci city and MCC. As a result we got Performance Evaluation, Effect of Using Label Power-set Encoding and then we presented Comparison with Other Methods.

In the concluding section before references, we provided a brief Summary with our study's Limitations and Future Work.

Chapter 2

Background

2.1 Biological Background

The organic molecules of a biological cells are also the unique constituents of cells. Maximum of the organic compounds can be roughly divided into four classes of molecules. Those divisions are as follows: nucleic acids, lipids, carbohydrates and proteins. Among these nucleic acids, proteins, and most carbohydrates (the polysaccharides) are actually macromolecules. These macromolecules are known to be formed by the joining or more prominently known as polymerization of thousands of low-molecular-weight precursors. These particular precursors are amino acids, nucleotides and simple sugars respectively. There are other major components such as Lipids. Lipids are predominantly one of the major component of cells. The remainder of the cell mass is composed of a variety of minor organic molecules with mentioned macromolecular precursors.

All the organic molecules has specific tasks to accomplish in a cell. Nucleic acids convey genetic information of the cell. The primary responsibilities of proteins include executing the tasks directed by that genetic information carried by nucleic acids. In the cell, proteins are the most diverse of all macromolecules. Proteins aid the cell as structural components and also to the tissues. They also continuously acting their specific roles in the transportation and storage unit in smaller molecules. Each cell comprises several thousand different kinds of proteins. These vast amount of proteins accomplish a extensive variety and array of functions.

The example of the said transport work is that the proteins transport of oxygen by hemoglobin. Another task of protein is transmitting information between cells. For example, protein hormones ensure this particular work. Protein also famous for delivering a strong defense against various infection (e.g., antibodies). The most fundamental function or property of proteins is their capability to perform as enzymes. Enzymes are the catalyst of approximately all the chemical reactions in biological structures and systems. Therefore, proteins are in the role to direct virtually all activities of an organic cell. The dominant standing of proteins in organic chemistry is specified by their name too. The word Protein is derived from the Greek word Proteios, that carries the meaning of the first rank.

2.1.1 Proteins

There are many biologically significant macromolecules are present in an organic cell. Protein is one of those. Proteins play a prodigious role in prompting many responses and reactions that are biological. Proteins are basically polymers of 20 different amino acids. Each amino acid comprises of a carbon atom, also called the carbon. They are bonded to a carboxyl group (COO-), an amino group (NH3+), a hydrogen atom and a distinctively large side chain. The specific chemical properties of the different amino acid side chains control the roles of each amino acid in protein structure and function.

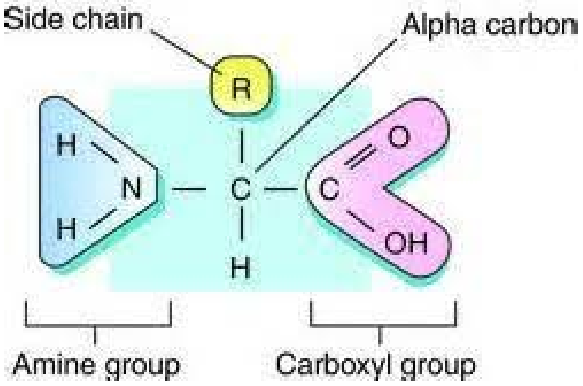


Figure 2.1: Structure of an amino acid

Protein, within its capacity, actively determines the structural composure of a cell along with its physical shape. Protein also empowers all proteins to carry out a certain work more precisely, that means protein also bring together other proteins for collective works. Protein play a crucial role to catalyze chemical reactions. Protein support cells in sending signals between cells and help mobilize other functions in a cell. In short, if we were to outline protein, we can concretely say that protein is actually a polymer of amino acids. This said polymer is linked well by a distinct bond called peptide bonds. There are three distinguished groups in an amino acid. They are:

1. Amino group
2. Carboxyl group
3. R group

The R group is actually a side chain by nature.

The status of R group can be quantified as such that R group actually governs the character of amino acid. Therefore, we are able to get the categories of amino acids by sorting out the character of it. As the categorization has been done already, as a result, we have found 20 main types out of amino acids.

Table 2.1: A list of the 20 standard amino acids

Amino Acid	**3 Letter**	**1 Letter**
Alanine	Ala	A
Tyrosine	Tyr	Y
Aspartic acid	Asp	D
Phenylalanine	Phe	F
Asparagine	Asn	N
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Proline	Pro	P
Valine	Val	V
Cysteine	Cys	C
Glutamic acid	Glu	E
Methionine	Met	M
Isoleucine	Ile	I
Leucine	Leu	L
Glutamine	Gln	Q
Glycine	Gly	G
Histidine	His	H
Lysine	Lys	K
Arginine	Arg	R

The structure of protein is remarkably fascinating in many ways. If we carefully observe the three-dimensional structure of a protein, it will provide ample insight about the core and fundamental functions of a protein. And as described earlier, the structure of protein is considerably influenced by amino acids. More precisely, the linear sequence of amino acid help us to sort out the three dimensional structure of a protein.

Renowned scientist Afinsen has earlier discovered proteins remarkable capacity to fold into proteins native structure. By his opinion, proteins do it spontaneously too. The studies have found ample evidence that has led to the conclusion that amino acid sequence governs the said folding event of proteins. It is absolutely important to comprehend the point that of how proteins fold in space. It has a genuine and strong possibility to reveal how proteins actually react in various biological conditions and different circumstances.

Computational biology is trying to break into the vast array of discoveries that are absolutely needed very quickly for the modern science to improve rapidly. Modern artificial intelligence and big data projects are eyeing the opportunity to work with biology. The sheer amount of data stored in genes and the mammoth task done by a single organism may potentially change the face of this civilization. The synchronization of all possible modern science is eventually being led to bio-informatics for its vastness and opportunity. Thus, computational biology has a large scope of working with the problem of determining the structure of a protein by its sequence.

We can fragment or categorize the protein structure into a hierarchy due to their amino acid chain. The categories can be:

content...

1. Primary
2. Secondary
3. Tertiary and
4. Quaternary

To demonstrate a strong and clear instance, we have chosen a relatively short protein. With this selected one we will be able to see primary, secondary and tertiary structures of a protein.

$$ELYSALANKCCHVGCTKRSLARFC$$

That is an example of primary protein sequence. This sequence can be seen in Human Relaxin, 6rlxA. This consists of a sequence of letters. Each letter here represents an amino acid.

The secondary structure consists of the alpha helix, beta sheet and beta turn. For a short segments of protein, the secondary structure is a better stable system for it to sustain. 6rlxAs secondary structure can be seen below (taken from PDB):

$$ELYSALANKCCHVGCTKRSLARFC$$
$$HHHHHHHHHHHTEEHHHHHTT$$

Here H is an alpha helix, E is a beta strand, T is a turn and an empty spot points out that there is no secondary structure has been assigned.

The overall conformation of a ploy-peptide chain is referred by the tertiary structure. Tertiary structure of the protein is greatly influenced by chemical properties of individual amino acid side chains. For protein 6rlxA, the tertiary structure is shown in figure 2.2 The figure 2.2 showing a short, relatively simple protein.

2.2

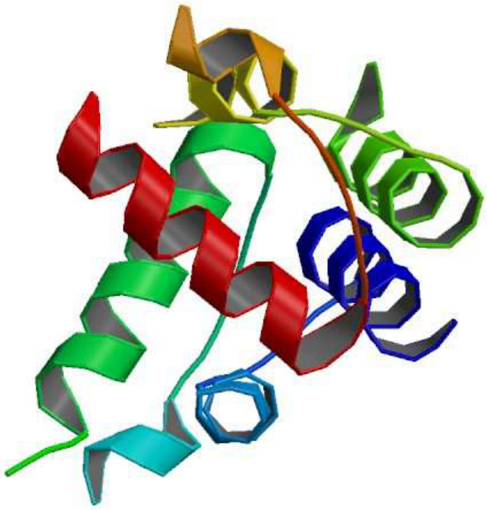


Figure 2.2: 3D structure of 6rlxA (taken from PDB)

Structure of proteins are very complex and varied in different ways. In some cases, proteins can be way more longer with extremely complex tertiary structure within them. Moreover, some proteins have multiple protein subunits. Each subunit of that protein has its own tertiary structure. This fascinating case or the arrangement by which the subunits assemble in a protein is called quaternary structure. 2.2

Multiple methods are developed to identify and define each structure of a protein. Here we are presenting an overview of the most used ways of determining protein structure:

1. Genetic sequencing of biological element such as proteins projects rapidly produce new protein sequences. There is a challenge about it. The challenge is that the number of sequences with known 3D structure surges much less progressively. The reason behind it can be stated as that the laboratory methods for determining protein structure are luxurious and inefficient (as it consumes relatively more time). Among the laboratory method Nuclear Magnetic Resonance (NMR) and X-ray crystallography are the perfect example of the statement mentioned earlier. To illustrate this point, we can present a simple statistical fact. The Universal Protein Resource (UniProt) contains about 8,000,000 protein sequences in compared to the number of protein structures stored in the Protein Data Bank (PDB) is around 95,000.

2. Another method is X-ray Crystallography. This method includes the process where an X-ray is fired upon a crystal. First some proteins are needed to be purified and concentrated to form a crystal. This crystalized protein than fired upon by an X-ray. Due to X-ray fire, there will be some diffraction. Next, the diffraction pattern is measured and upon that measurement a 3D model picture is generated to determine the protein structure type. However, this incredibly sophisticated and complex process has various challenges that takes considerable time and effort to complete. Also, some type of proteins are very difficult to crystalize even with this process.

3. Nuclear Magnetic Resonance Spectroscopy or in short NMR Spectroscopy is another distinctive method of determining structure of a protein. The application of the process is very limited and applied for only shorter kind of protein. In this process, first proteins are submerged in a special solution of water. Then the sample is placed under a magnetic field. After that different spectrum of radio waves are sent through, where the protein in sample absorbs the radio wave. The prominent feature of the system is that different proteins different atomic nuclei absorbs different frequencies of the radio waves. Finally, based on the absorption and some other characteristics the protein structure is determined.

4. All the experimental methods are relatively difficult to perform. Also, they have various challenges that are expensive and inefficient in terms

of time that takes to conduct the experiments. Thus, methods that are involved with computational approach are much more desirable in some ways. Ab Initio introduced a 3D structure of a protein which is beautifully extracted from physical principles. His methods regarding protein structure prediction is a brand-new technique. This technique is actually based on famous Ramachandran Plots. This method, unfortunately is by far proven to be impractical in a special case. That is, this method is not practical for large proteins and high-resolution models. It is due to the grand search space.

5. There is another method called Homology. This method is based to find proteins with already explored structure with similar sequences to a protein which possesses unknown structure. The assumption was made that the structure of a homologous protein assumed to be similar and more can be added that it can be cast-off or consider as starting point for conjecturing the structure of a new biological cell. However, proteins which show sequential similarities minimum 15 percent can potentially have similar tertiary structure. And homology based tactics are decided to be inappropriate for these cases.

6. Protein Threading is an interesting approach. It is an eloquent method of determining the fold of a protein. It does so by comparing with a set of templates. Quite interestingly, protein threading and homology based methods are very analogous. However, methods that are based on homology are restricted to proteins that have high sequential similarity. On the other hand, the Protein Threading is generally used for such proteins which has lower sequential similarity. Popular Threading methods include HHsearch and SPARKS-X.

2.1.2 Position Specific Score Matrices (PSSM)

PSSM or Position-Specific Scoring Matrix is one of a kind of scoring matrix. It is frequently deployed in case of protein BLAST searches. In those searches amino acid substitution scores are given separately for each position in a protein multiple sequence alignment. PSIBLAST (Position-Specific Iterative Basic Local Alignment Search Tool) derives a position-specific scoring matrix (PSSM). This PSSM form the multiple sequence alignment of sequences detected above a given score threshold using protein BLAST.

BLAST (Basic Local Alignment Search Tool) is can be iterated as a sequence similarity search method. This method learns about the given protein and then compares that protein to the set of protein sequences provided in a target database. The goal is to identify regions of local alignment and also report those alignments that score above a given score threshold. The BLAST primarily finds regions of local similarity between sequences. The program comparatively relates protein sequences to sequence databases and simultaneously calculates the statistical significance of those matches.

To infer on functional and evolutionary relationships between sequences, BLAST can be a useful tool. It can certainly assist to identify members of gene families. PSI-BLAST utilizes the outcomes of BLAST run for several iterations. Here PSI-BLAST uses the best matches from each iteration in the next iteration. The PSSM captivates the conservation pattern in alignment. Then stores it as a matrix of scores for each position in the alignment. Here, a highly conserved positions get high scores and similarly, weakly conserved positions receive scores nearing zero.

PSSM scores normally revealed as positive or otherwise negative integers. Sometimes, by chance, the given amino acid substitution happens more frequently in the alignment than anticipated. Positive scores of PSSM indicates those cases. Negative scores point out that the substitution occurs less frequently than anticipated by the researchers. There can be large positive scores. It often indicate critical functional residues. It can be either active site residues or residues that are someway required for other intermolecular interactions.

PSI-BLAST works in a very neat way. For each input protein, it returns 2 separate matrices. Given L is the length of the input protein, each matrix is L 20 in size. Here, both of these matrices store substitution probabilities. One of them holds linear probabilities and next contains log-odds within them.

2.2 Machine Learning Background

Machine learning is a fascinating arena of artificial intelligence (AI) research. It is a subfield of computer science where we try to find out how computer or modern computing tools can be used in advancing the perception, increase efficiency in cognition and improve timing and feasible complex action to experience accomplishing multi layered tasks. Arthur Samuel in the year 1959 has cleverly described machine learning as "Computers ability to learn without being explicitly programmed". Machine learning is gradually transformed and matured from the study of pattern recognition and computational learning theory in artificial intelligence. Machine learning reconnoiters the learning and edifice of algorithms that can acquire from and make predictions on large and complex dataset. Such algorithms transcend following strictly inert program instructions by making data-driven decisions by building model from sample inputs.

There are numerous things we want to analyze, work on and get to have insight about. But not all of it can be explicitly programmed. Some decisions are to be taken depended on the available data in the middle of the analysis. This intelligence is much more demanded and extraordinarily needed in todays reality. Therefore, machine learning is very important. In plain terms, machine learning is the discipline of getting computers to perform without being overtly automated. This is usually employed in a series of computing tasks. These tasks are oriented on designing and programming explicit algorithms. Algorithms as such where good performance is difficult or infeasible. We can provide viable example as email filtering, detection of network intruders and the incident where malicious insiders working towards a data breach. Optical character recognition (OCR), learning to rank and computer vision etc can be presented as examples too.

Two types of machine learning algorithms are mainstream. One is supervised learning and another is unsupervised learning. Inferring a function from labeled training data is called Supervised Learning. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). This structured method is used in many scientific cases and conducting scientific studies to find a particular data or solution to a well-studied problem or problem sets.

On the other hand unsupervised machine learning is somewhat opposite of it. Instead of setting any example or providing supervisory signal, it relies on inference. This method incorporates machine learning task of inferring a function to describe hidden structure from "unlabeled" data. Here unlabeled data means that in the given data or observations, there are no classification or categorization. The dataset constructed for the purpose of experiment here in our paper are labeled data set. Therefore, according to the classification mentioned above, we have tried supervised learning algorithms.

A number of supervised learning algorithms are used to in our branch of study. Among those some are more notable than the others. Linear Regression, Logistic Regression, Decision Tree, Support Vector Machine (SVM), Naive Bayes, KNN, K-Means, Random Forest etc. are most prominent among the supervised learning algorithms. It has been shown in previous literatures that support vector machine (SVM) has the capacity to provide most promising result in protein sub-cellular localization area. Thus, keeping that point in mind, in this research we have adopted SVM as our classifier.

2.2.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) has widespread use in Bioinformatics and considerably performed better then other classifiers. SVM has secured promising results for protein sub-cellular localization in various studies. SVM make an effort to lessen the rate of error in prediction by finding the hyperplane that provides the largest margin based on the concept of support vector theory. It transforms the provided input data to higher dimensions using the kernel function to be able to find support vectors (for non linear cases). The classification of some known points in input space x_i is y_i which is defined to be either -1 or $+1$. If x' is a point in input space with unknown classification then:

$$y' = \sin(\sum_{i=1}^{n} a_i y_i K(x_i, x') + b)$$

where y' is the predicted class of point x'. The function $K()$ is the kernel function, n is the number of support vectors and a_i are adjustable weights and b is the bias. We tested the effect of our proposed method on three base classifiers: Decision Tree (DT), Random Forest (RF) and Support Vector Machines (SVM). Among these three classifiers, SVM performed best. Support vector machines [9] are classifiers that tries to separate the different classes in the dataset using a hyperplane learned from the training data that maximizes the separation between the borderline instances. it is also known as the maximum margin classifier. SVMs generally tries to optimize a multiplier function that goes as the following:

$$L = \underset{\alpha}{argmax} \sum_{j} \alpha_j - \frac{1}{2} \sum_{j,k} \alpha_j \alpha_k y_{)} j y_k \phi(\vec{x}_j.\vec{x}_k)$$

The prediction of a SVM classifier is defined as below:

$$h(\vec{x}) = sign(\sum_{j} \alpha_j y_j (\vec{x}.\vec{x}_j) - b)$$

Here the transformation of the data points by the function ϕ could be linear, polynomial or any other kernel functions. Multi-class SVMs are extension of binary SVMs with an appropriate function (eg. *softmax*) to approximate a multinoulli distribution. The parameters used for SVM in our experiments were, gamma (γ) = 0.05 and $C = 3000$ and radial basis function (RBF) kernel was used.

Therefore, SVM is well-thought-out to be one of the finest pattern recognition techniques among the many methods that are at disposal of the researchers SVM is extensively put into conduct studies in Bioinformatics. Till date it has outdid other classifiers and attained very promising results for protein sub-cellular localization.

2.2.2 Decision Tree

Decision tree classification is based on selecting attributes as decision nodes in each level of the tree where instances are divided based on the attribute value or decision. Generally, nodes are mapped to decisions based on the values of the attribute that can discriminate the instances best. As discriminatory information, gini impurity, entropy and information gain are widely used. For a better generalization on the training dataset often trees are pruned.

Let us begin from the top by answering what is decision tree learning. Decision tree learning is a frequently used machine learning algorithm. The advantages of using decision trees include its simplicity and being straight forward. It is fairly very easy to grasp and can be easily explained to the humans. Decision trees deliver a way to estimate discrete valued functions. It is also fairly robust to work with particularly noisy data. Decision trees can be represented using the typical Tree Data Structure. Decision tree learning primarily make use of a decision tree to go from understanding or taking observations about an item in consideration to conclusions about the items target value. It is one of the most used predictive modelling approaches used in statistics, data mining and machine learning.

In decision tree learning, a decision tree - now known by the umbrella term CART (Classification and Regression Tree) - can be used to visually. It is also can explicitly represent decisions and decision making from all types of data given to it. Though, it is common to use a tree-like model for decisions, learned trees can also be represented as sets of if-else-then rules time to time. Decision trees though can be utilized for both classification and regression in analysis and modeling. But decision tree is primarily used for classification. Lets visit a peak at how a classic representation of a decision tree can be perceived. Decision trees perform classification after sorting the given instances in a top-down approach. That is, in short from the root to the leaf. Each non-leaf node will split the set of instances. It will be based on a test of an attribute. Where each branch emanating from a node resembles to one of the conceivable values of the said attribute in the node of the tree. The leaves of the decision tree stipulates the label or the class. Within this class, a given instance belongs.

Decision Trees represent a disjunction of conjunctions of constraints on attributes values of instances. That is, Decision Trees represent a bunch of AND statements chained by OR statements. A valid concern can be expressed as when should one use a decision tree?. Well, it is an interesting concern to address. A fairly simple answer would be that a decision tree must be used when it is imperative for the humans to understand and communicate the mode! We can also take the below mentioned points not the count:

1. When you would like to produce minimalistic assumptions from the given dataset.
2. When you do not want to or do not have the patience or time to normalize the provided data.
3. When your precious dataset contains ample amount of serious noise (but not too much!).
4. When your data has the presence of skewed variables in the dataset.
5. When there are many and many missing attribute values can be found in the dataset.
6. When disjunctive descriptions are required
7. When you are in absolute need to build and test fast
8. When the dataset is fairly small in size

Well, now our discussion concerns about how is a decision tree usually built. Before we start classifying, we first need to build the tree from the available dataset. Most algorithms that have been developed for learning decision trees are variations of the core algorithm that employs a top down, greedy search through the possible space of decision trees. As there is some aspects of decision tree that needs to be careful with. That is Overfitting In A Decision Tree. When given dataset becomes larger, the decision tree is in its usual process tends to become longer. In those cases, noise and corrupt/incorrect data can have a disadvantageous influence on the decision tree. This results in the decision tree overfitting the dataset. That means, decision tree performs satisfactory for the training data. But ultimately fails to yield an appropriate approximation of the target perception when it come across the actual data. Overfitting can also occur when insufficient data is supplied to build the decision tree. In order overcome the overfitting scenario, one of the following two things shall be done or keep in mind to do. Either the decision tree should stop growing before it overfits the data or an overfitting tree should be pruned to reduce the error!

2.2.3 Random Forest

Random Forest classifier is an ensemble classifier with decision trees as base classifiers. It is an application of bootstrap aggregating or bagging technique that first samples the original dataset into a number of datasets by a sampling with replacement technique. This bootstrap method of sampling can radically reduce noise or outliers in the data. After creating K samples from the original dataset, K decision trees are learned on each dataset and the decision of these bag of classifiers are combined by taking voting on the predictions made by them. The decision trees learned by the random forest algorithm are random in nature. In each iteration, features are selected randomly and based on the selected features only the trees are learned. The underlying decision tree algorithm uses discriminatory information to build the tree structure of the data for classification.

Lets look into the Random Forests in detail. We will though start with the simple statement. That is, Random Forests are used as a Method to Reduce Variance. Decision Trees as we discussed in previous section are reputed for showing high variance and low bias. This is mostly because of their ability to model complex relationships. Even they can model complex relationship to the point of overfitting the noise in the data. Here overfitting means not being general enough or unusual data. Simply putting into words: Decision Trees train models that are usually accurate. But that said scenario often show a large degree of variability between various data samples taken from the exact same dataset!

There comes the Random Forests, as it reduce the variance that can cause errors in Decision Trees by aggregating the different outputs of the individual Decision Trees. We can find the average output given by most of the individual Trees through majority voting. Therefore, it does the work of smoothing out the variance so that the model will be less prone to producing results far away from the actual or real values. The impression behind Random Forests is to accept a set of high-variance, low-bias Decision Trees and then convert them into a new model which has low variance as well as low bias.The next logical query is that why the Random Forests are actually random? The random in Random Forest originates from the fact that the algorithm that trains each individual decision tree with different subsets of the training data. And each node of each decision tree is potentially split using a randomly selected attribute from the given data. The algorithm is able to create models that are no way correlated with each other by including the element of randomness.

Due to this fact, something good happens. That is, the possible errors will spread out evenly throughout the model. This also means that they will eventually be canceled out through the majority voting decision strategy of Random Forest models.There is also a concern that How Would a Random Forest Work in the Real World? The query is valid and can be answered easily. Imagine that you are bored of seeing the same science fiction movie repeatedly. Now, you dreadfully crave to find a new movie that you may like. So, eventually, you go online to find good recommendations from like-minded people or peers. As you are browsing through, you find a website that lets real people provide you science fiction movie recommendations based on your likings. So how does it work? First, to avoid recommendations that are simply random, you would fill out a questionnaire about your basic science fiction movie preferences. You will also provide a baseline for the type of science fiction movies you usually watch. With that information, folks from the website start to scrutinize science fiction movies using the criteria (features) that you provided.

Each individual is fundamentally at work as a decision tree. Individually, the people making suggestions are pretty likely to generalize your science fiction movie preferences poorly. For example, one person may conclude that you do not like any science fiction movie from before the 1980s, and will therefore not include any in your recommendations. However, in all fairness, this could be an inaccurate assumption. It would cause you to not receive suggestions for science fiction movie you are likely to enjoy. Now, the question is, why is this mistake happening? Each of the people giving recommendations only has limited information about your preferences. Also, they are primarily biased by their own individual taste in science fiction movies. To be able to fix this, we would be needing to combine the suggestions from many individuals (each acting as a Decision Tree) and use majority voting on their suggestions (essentially creating a Random Forest).

But, there is still one more problem remains because each of the people is using the same data from the same questionnaire, the resulting suggestions will not be varied and may be highly biased and correlated! To expand the range of suggestions that may come into your way, each of the recommenders is given a random set of your answers instead of all of them, meaning that they have less criteria with which to make their recommendations. In the end, the extreme outliers are eliminated through majority voting, and you are left with an accurate and varied list of recommended science fiction movie.

Chapter 3

Materials and Methods

3.1 Methodologies

In this section, we describe the details of the method and materials used in this paper. A system diagram of our proposed model is shown in Figure 3.1. Our system starts by fetching the protein sequences in the dataset and feeding them into PSI-BLAST [1] software to fetch Position Specific Scoring Matrix (PSSM) files using the nr database.

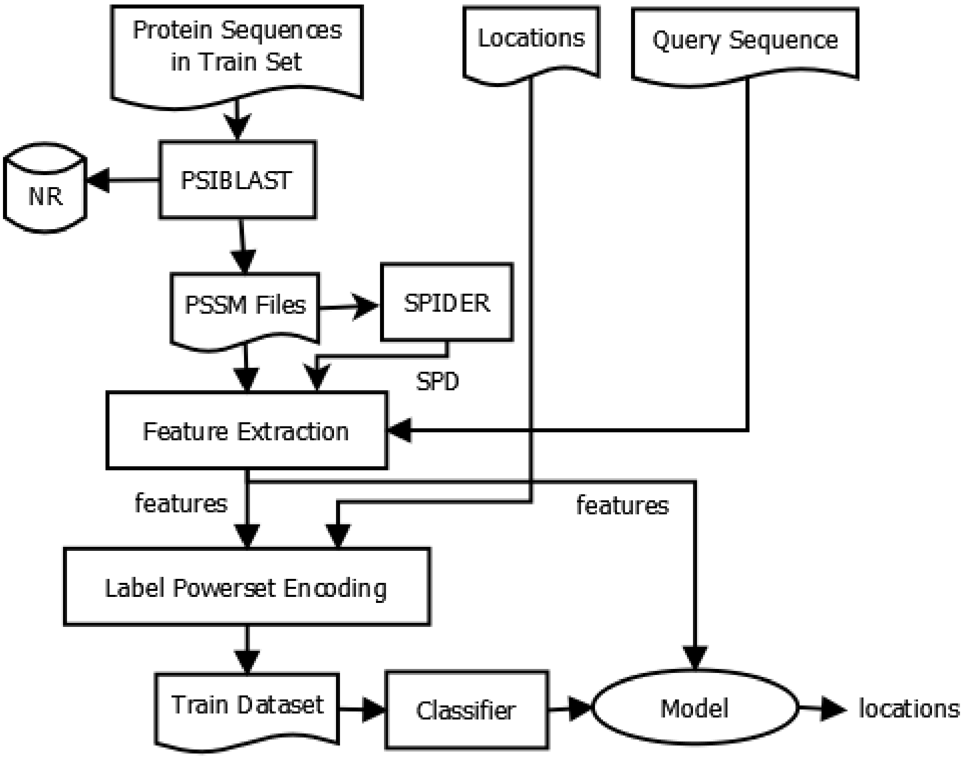


Figure 3.1: System Diagram

PSSM files are then fed to the SPIDER [22] software to generate secondary structure related information prediction to generate SPD files. SPD and PSSM files are then used to generate features for classification using a feature extraction procedure. Extracted features and locations (multiple or single) of the proteins are then fed into a label space transformation module that converts the label space into a new one. The training dataset with newly transformed labels are then fed into a classifier to learn a model, which stored later can be used to find locations for any query sequence. Rest of the section follows the general suggestions made in [5].

3.2 Data Set

In this research we have used two dataset which have been used widely among this field of literature [23], [8], [7], [30], [25] for Gram-negative subcellular localizations. The details of this two dataset are described below:

3.2.1 Gram-Negative Bacteria Protein Dataset

For gram-negative sub-cellular localizations we have used dataset that was introduced in the literature [25], [8], [7], [6]. This dataset contains total 1456 protein samples which belongs to eight gram negative sub-cellular localizations. Among this 1456 samples there are total 1392 different protein sample. Among 1392 proteins there are total 1328 protein samples which belongs to only one or single location while the rest 64 protein samples belongs to two location. Thus gram-negative bacteria protein dataset contains total 1456 (1328 + 64 * 2) protein samples. The eight locations name and total number of protein samples each location contains are shown at Table 3.2. This dataset is available at the web-link http://www.csbio.sjtu.edu.cn/bioinf/Gneg-multi.

Table 3.1: Details of gram-negative bacteria protein dataset

No.	Sub-cellular location	Total protein samples
1	Cell inner membrane	557
2	Cytoplasm	410
3	Cell outer membrane	124
4	Extracellular	133
5	Periplasm	180
6	Fimbrium	32
7	Nucleoid	8
8	Flagellum	12
Total number of locative proteins		1456
Total number of different proteins		1392

For classifying multi location proteins we have used the same method which was used in the literature [23], [25]. In gram positive and gram negative bacteria protein dataset which samples are in multi location means more than one location, we have used those multi labeled proteins as several single labeled protein samples based on their labels or classes which they belongs to. For example a protein sample which has two label or class location, we have used them in our experiment as two single labeled protein sample. Then we have performed our experimental classification task. Thus by making and adding extra protein samples, we have calculated the possible worst case scenario as we might not be able to predict the multi labeled protein samples or the extra label of a single protein that has more than one label. In this way, predicting a protein's sub-cellular location, we guarantee that we have considered all the worst case that can occur in performing my prediction task.

3.2.2 Plant protein dataset

The presented plant dataset has created from Swiss-Prot 55.3. This dataset has 978 plant proteins and they are distributed in 12 locations (see Table2(b)). When we closely examine as to how these proteins are distributed, it reveals some fascinating results. Out of mentioned 978 plant proteins, 904 of them belong to one sub-cellular locations. Another 71 is found to be belongs to two locations. There are 3 proteins that belongs to three locations and none to four or more locations. To put this into perspective, we can say, only 8 percent of the plant proteins in the presented dataset are located in multiple locations. The sequence identity of this dataset was cut off at 25 percent.

3.3 Normalization of PSSM

In our experiment, we have made two main groups of PSSM matrix named as Normalized PSSM matrix and Non-Normalized PSSM matrix. Non-Normalized PSSM is the exact PSSM matrix which is the output matrix of PSSIBLAST software. Normalized PSSM matrix was used in the literature [29]. According to this [23] literature PSSM matrix can be represented as:

$$P = \begin{bmatrix} U_{1,1} & U_{1,2} & \cdots & U_{1,19} & U_{1,20} \\ U_{2,1} & U_{2,2} & \cdots & U_{2,19} & U_{2,20} \\ . & . & \cdots & . & . \\ . & . & \cdots & . & . \\ . & . & \cdots & . & . \\ U_{L,1} & U_{L,2} & \cdots & U_{L,19} & U_{L,20} \end{bmatrix}$$

The size of the PSSM matrix is $L \times 20$, here L is the length of the amino

Table 3.2: Details of the new plant dataset

No.	Sub-cellular location	Total protein samples
1	Chloroplast	286
2	Cytoplasm	182
3	Mitochondrion	150
4	Nucleus	152
5	Cell membrane	56
6	Vacuole	52
7	Golgi apparatus	21
8	Endoplasmic reticulum	42
9	Cell wall	32
10	Plastid	39
11	Peroxisome	21
12	Extracellular	22
Total number of different proteins		978
Total number of locative proteins		1055

acid sequence or simply primary protein sequence, $U_{i,j}$ represents the score of amino acid which is located at the i-th location of the protein sequence which is changed into amino acid j during the process of evolution. In order to make the normalization of PSSM matrix, we have computed and formulated a new PSSM matrix N using the information from original PSSM matrix P. We are refering this new matrix N as my new normalized PSSM in this paper. The normalized matrix N is computed as follows:

$$N = \begin{bmatrix} V_{1,1} & V_{1,2} & \cdots & V_{1,19} & V_{1,20} \\ V_{2,1} & V_{2,2} & \cdots & V_{2,19} & V_{2,20} \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ V_{L,1} & V_{L,2} & \cdots & V_{L,19} & V_{L,20} \end{bmatrix}$$

where, $V_{i,j} = \dfrac{U_{i,j} - Z_y}{Z_x - Z_y}$; i = 1,2,.....,L; $Z_x = max(P)$ & $Z_y = min(P)$

We have normalized all PSSM matrix one by one. First we have took one PSSM file and find out the highest score Z_x and lowest score Z_y of the matrix. Then putting this maximum and minimum value in my formula we have calculated the normalized score. Algorithm for converting a PSSM matrix to its corresponding Normalized PSSM matrix is shown at Algorithm 1.

From Normalized PSSM matrix we have extracted 7 features:

1. PSSM-C Composition of Normalized PSSM (Feature vector size 20)

Algorithm 1: Normalization of PSSM Matrix

```
$Z_x \leftarrow$ P[0,0];
$Z_y \leftarrow$ P[0,0];
$L \leftarrow$ Length of PSSM Matrix;
$P \leftarrow$ Original PSSM Matrix;
$V \leftarrow$ Empty Array of Size $L \times 20$;
for $i = 0;\ i < L;\ i = i + 1$ do
    for $j = 0;\ j < 20;\ j = j + 1$ do
        if P[i,j] > Z_x then
            $Z_x \leftarrow$ P[i,j];
        if P[i,j] < Z_y then
            $Z_y \leftarrow$ P[i,j];
for $i = 0;\ i < L;\ i = i + 1$ do
    for $j = 0;\ j < 20;\ j = j + 1$ do
        $V_{i,j} = \frac{P_{i,j} - Z_y}{Z_x - Z_y}$;
```

2. PSSM-SD (Feature vector size 80)
3. PSSM-SAC (Feature vector size 100)
4. PSSM-AC from Normalized PSSM Auto-Covariance
5. One-lead Bi-gram of Normalized PSSM (Feature vector size 400)
6. Torsional Angles Composition
7. Auto Covariance of Probabilities from Normalized PSSM (Feature vector size 200)

3.4 Feature Extraction Method

Feature extraction is fairly simple to describe and understand. It is a process, and this involves keeping information relevant to the classification task. And also discarding other irrelevant information. Feature extraction includes lessening the amount of resources necessary to describe a vast set of data. Number of variables involved is always a challenge to be managed when performing analysis of complex data. Analysis with a large number of variables generally requires a large amount of memory and computation power. Besides that, also it may cause a classification algorithm to overfit to training samples. Eventually it can generalize poorly to new samples.

Feature extraction has the capacity to describing the data with sufficient accuracy. Because, it is method of constructing combinations of the variables to get around the mentioned problems. Since most classifiers only accept fixed length feature vectors, the feature extraction step is also a way of creating a fixed length feature vector from the variable length protein sequences.

We use evolutionary information fetched by PSSM files to extract features. PSSM files contain substitution probabilities of each amino-acid residue at each position of the given protein sequence. Its a matrix, P of dimension $L \times 20$, where L is the length of the protein. First, this matrix is normalized using a method similar to that described in [29] and found to be effective for subcellular localization previously. Lets, call this matrix, N with same dimension as P. Now features are extracted from this matrix.

3.4.1 Composition Feature

This feature is extracted from both PSSM matrix and Spider SPD3 matrix. To calculate this feature we have taken one column by one column at a time from the respective matrix and summed up all the rows value of this particular column and finally divided it with the length of the protein. The equation for this feature is given below:

$$Composition_j = \frac{1}{L} \sum_{i=1}^{L} N_{i,j}$$

Here N is the corresponding matrix, L is the protein length and j is the respective column. The dimensionality of this feature vector will be $(Number\ of\ columns)$. Algorithm for extracting composition feature is shown at Algorithm 2.

Algorithm 2: Composition Feature Extraction

```
$N \leftarrow$ Matrix from which feature will be extracted;
$L \leftarrow$ Length of the Protein;
$C \leftarrow$ Number of matrix column;
$V \leftarrow$ Empty array of size $C$;
for $j = 0;\ j < C;\ j = j + 1$ do
    $sum \leftarrow 0$;
    for $i = 0;\ i < L;\ i = i + 1$ do
        $sum = sum + N_{i,j}$;
    $V_j = \frac{sum}{L}$;
```

We have extracted total 8 composition features from both PSSM matrix and Spider SPD3 matrix. Using composition feature extraction method we have extracted 2 features from PSSM matrix:

1. Composition of PSSM (Feature vector size 20)
2. Composition of Normalized PSSM (Feature vector size 20)

3.4.2 PSSM-SD Feature

This method is specifically proposed to add more local discriminatory information about how the amino acids, based on their substitution probabilities (extracted from PSSM), are distributed along the protein sequence [13]. We propose this segmentation method in the manner where segments of a protein sequence are of unequal lengths and each segment is represented by a distribution feature which is computed as follows.

Algorithm 3: PSSM-SD Feature Extraction

```
$N \leftarrow$ PSSM Matrix;
$L \leftarrow$ Length of the Protein;
$C \leftarrow$ Number of matrix column;
$F_p \leftarrow$ Desired value of $F_p$, e.g 5, 10, 25;
$V \leftarrow$ Empty array of size $(100 \div F_p) \times C$;
$k \leftarrow 0$ ;
for $j = 0;\ j < C;\ j = j + 1$ do
    $T_j \leftarrow$ Sum of jth column;
    $partialSum \leftarrow$ 0;
    $i \leftarrow$ 0;
    for $tp = F_p;\ tp <= 50;\ tp = tp + F_p$ do
        while $partialSum <= tp \times (T_j \div 100)$ do
            $partialSum = partialSum + N_{i,j}$;
            $i = i + 1$;
        $V_k = i$;
        $k = k + 1$;
    $partialSum \leftarrow$ 0;
    $i \leftarrow$ L;
    $index \leftarrow$ 0;
    for $tp = F_p;\ tp <= 50;\ tp = tp + F_p$ do
        while $partialSum <= tp \times (T_j \div 100)$ do
            $partialSum = partialSum + N_{i,j}$;
            $i = i - 1$;
            $index = index + 1$;
        $V_k = index$;
        $k = k + 1$;
```

First, for the jth column in PSSM, we calculate the total substitution probability $T_j = \sum_{i=1}^{L} P_i j$. Then, starting from the first row of PSSM, we calculate the partial sum S_1 of the substitution probabilities of the first i amino acids until reaching $F_p\%$ of the total sum $S_1 = \sum_{i=1}^{I_j^1} P_i j$. Using the distribution factor $F_p\%$, we calculate the I_j^1, where I_j^1 corresponds to the number of the amino acids such that the summation of their substitution probabilities is less than or equal to the $F_p\%$ of T_j. Similarly, we continue to calculate the partial sum of the first i amino acids (starting from the first row of PSSM) until reaching $n \times F_p\% = 50\%$ of the total sum ($S_n = \sum_{i=1}^{I_j^n} P_i j$) and calculate the I_j^n corresponding to the number of amino acids such that the summation of their substitution probabilities is less than or equal to 50% of T_j. Therefore, starting from the first row of PSSM, we extract n features

$()I_j^1, I_j^2, ..., I_j^n)$ corresponding to the number of segments until reaching 50% of T_j.

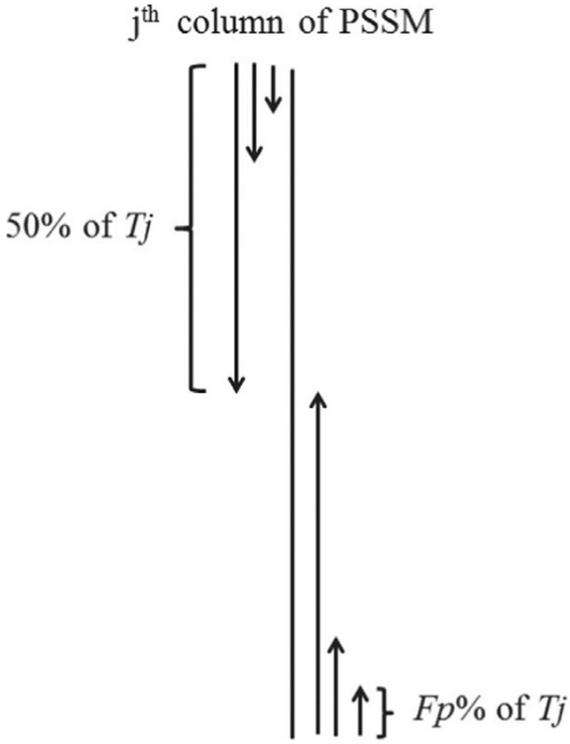


Figure 3.2: The segmentation method used to extract PSSM-SD feature group from the jth column of PSSM

We repeat the same process beginning from the last row of PSSM for the jth column. We calculate the partial sum of the substitution probabilities of the first i amino acids until reaching $n \times Fp\% = 50\%$ of the total sum which is $S_{n+1} = \sum_{i=1}^{I_j^{n+1}} P_ij$ until reaching $S_2n = \sum_{i=1}^{I_j^2n} P_ij$, respectively and calculate I_j^{n+1} until reaching $I_j^2n.I_j^{n+1}$ and I_j^2n correspond to the number of amino acids such that the summation of their substitution probabilities are less than or equal to $Fp\%$ and $n \times Fp\% = 50\%$ of T_j, respectively (starting from the last row of PSSM). Therefore, starting from the last row of the PSSM, we extract n features $(I_j^{n+1}, I_j^{n+2}, ..., I_j^{2n})$ corresponding to the number of segments until reaching 50% of T_j. In this manner we extract $2n$ segmented distribution features for each column in PSSM. The method used to calculat ePSSM-SD feature group from the jth column of PSSM is shown in Fig. 3.2. We repeat the same process for all 20 columns corresponding to 20 amino acids in PSSM.

In this study, we adopt three values for F_p (5, 10, and 25)to investigate the effectiveness of the number of segments on the achieved results and find

the suitable number of segments to explore local discriminatory information embedded in PSSM. We have used other choices for F_p but these three remains the best representatives of all the choices. To maintain the generality and simplicity of the segmentation method, we avoid a very specific segmentation method as it might not be applicable for all cases. For PSSM-SD feature group, using $Fp = 5$, we divide the protein sequence into 20 segments ($n = 10$from each side) and extract 400 features in total in this feature group ($20 \times 20 = 400$). Similarly, using $F_p = 10$ ($n = 5$ from each side) we divide the protein sequence into 10 segments and extract 200 features in total ($10 \times 20 = 200$) and by using $F_p = 25$ ($n = 2$ from each side), we extract 80 features in total ($4 \times 20 = 80$). General formula for feature vector size of PSSM-SD is
$(100 \div F_p) \times (Number\ of\ columns\ in\ the\ matrix)$.

However, in literature [10] has proven that $F_p = 25$ gives the best result and we also investigate the same result, so in our final experiment we have adopt $F_p = 25$ which gives 80 ($(100 \div 25) \times 20 = 80$) as feature vector size. Thus in this report we have only mentioned the feature vector size of PSSM-SD as 80. Algorithm for extracting PSSM-SD feature is shown at Algorithm 3.

3.4.3 PSSM-SAC Feature

This feature was introduced in the literature [10]. It was shown that information about the interaction of neighboring aminoacids along the protein sequence can play an important role in providing significant local discriminatory information and enhancing protein subcellular localization prediction accuracy [3], [28]. To extract this information, the concept of auto covariance has been widely used in the literature in different ways (e.g.bi-gram([28]), tri-gram ([26]), auto correlation ([14], [12])). Among all these methods, pseudo amino acids composition has attained the best results to extract local information ([8], [31], [19]). In the present study, we extend the concept of segmented distribution features as described in the previous subsection to compute the auto covariance features from the segmented protein sequence. This is done to enforce local discriminatory information extracted from PSSM.

To extract this feature group, we calculate the auto covariance of the substitution probabilities of the amino acids using K_p as the distance factor in the following manner. Starting from the first row of PSSM, for the jth column of PSSM, we calculate K_p auto covariance features for the first I_j^1. Similarly, we calculate auto covariance until reaching the first I_j^n amino acids. Then starting from the last row of PSSM for the jth column of PSSM, We repeat the same process for I_j^{n+1} and until reaching $I_j^2 n$ (I_j^1 to I_j^n and I_j^{n+1} until reaching to $I_j^2 n$ are calculated in the way that is explained in the previous subsection). This process is repeated for all 20 columns of PSSM and corresponding features are calculated as follows:

$$PSSM - seg_{q,m,j} = \frac{1}{I_j^q - m} \sum_{i=1}^{I_j^q - m} (P_{i,j} - P_{ave,j})(P_{(i+m),j} - P_{ave,j}),$$

$$(q = 1, ..., 2n \;\&\; m = 1, ..., k_p \;\&\; j = 1, ..., 20)$$

Thus, we have extracted a total of $(nK_p + nK_p + K_p) = (2n+1)K_p)$ auto covariance features in this manner (for the jth column of the PSSM). For all 20 columns of the PSSM, segmented auto covariance of the substitution probabilities of the amino acids are extracted and combined to build the corresponding feature group which will be referred to as PSSM-SAC (PSSM-seg + PSSM-AC which consists of $20 \times (2n+1) \times K_p$ $features\ in\ total$).

In the literature [10] the authors have tried different values for K_p starting from 1 to 10 (1,2,3,.......,8,9,10). They have reported that $K_p = 10$ gives the best result for PSSM-SAC. So in our experiment we have used 10 as a value for K_p. Thus our feature vector size for PSSM-SAC is 100. General formula for feature vector size for PSSM-SAC is $K_p \times 20 \times 5$.

3.4.4 Auto Covariance Feature

A correlation factor coupling adjacent residues along the protein sequence [34] is known as Auto covariance (AC). It is also known as a kind of variant of auto cross covariance.

Algorithm 4: Auto Covariance Feature Extraction

```
$DF \leftarrow 10$;
$P \leftarrow$ Matrix from which feature will be extracted;
$L \leftarrow$ Length of the Protein;
$V \leftarrow$ Empty array of size $L\times$ (Number of matrix column);
$C \leftarrow$ Number of matrix column;
for $k = 0;\ k < DF;\ k = k + 1$ do
    for $j = 0;\ j < C;\ j = j + 1$ do
        $sum \leftarrow 0$;
        for $i = 0;\ i < L - k;\ i = i + 1$ do
            $sum = sum + P_{i,j} P_{i+k,j}$;
        $V_{k,j} = \frac{sum}{L}$;
```

It is a very powerful statistical tool which is used to analyze sequences of vectors [32], the Auto Covariance transformation has been widely applied in various fields of bioinformatics [20], [21], [17], [33], [35], [24]. Auto Covariance variables are able to avoid producing too many variants. The equation for this feature is given below:

$$AutoCovariance_{k,j} = \frac{1}{L}\sum_{i=1}^{L-k} N_{i,j}N_{i+k,j} \;\; (j = 1, ..., 20 \; and \; k = 1...DF)$$

where DF is the distance factor. Different values have been tested to find out the effective value of DF which gives the highest accuracy rate of prediction. In this research we have tested total 15 values for DF (DF = 1,2,3,4,.......,12,13,14,15) and took only one value which is DF = 10. We have observed that only DF = 10 gives the highest accuracy rate for my task. So, the effective value of DF is used as 10 for the employed benchmark since this value was investigated in other literature [11] which gives promising results for other benchmark datasets. The dimensionality of this feature vector will be $(Number\ of\ columns) \times DF$. Algorithm for extracting auto covariance feature is shown at Algorithm 4.

We have extracted total 8 auto covariance features from both PSSM matrix and Spider SPD3 matrix. Using auto covariance feature extraction method we have extracted 2 features from PSSM matrix:

1. Auto Covariance of Normalized PSSM (Feature vector size 200)

3.4.5 One-Lead Bi-Gram Feature

The equation for this feature is given below:

$$OneLeadBigram_{k,l} = \frac{1}{L}\sum_{i=1}^{L-2} N_{i,k}N_{i+2,l}$$

The dimensionality of this feature vector will be $(Number\ of\ columns) \times (Number\ of\ columns)$. Algorithm for extracting one-lead bi-gram feature is shown at Algorithm 5.

We have extracted total 8 one-lead bi-gram features from both PSSM matrix and Spider SPD3 matrix. Using one-lead bi-gram feature extraction method we have extracted 2 features from PSSM matrix:

1. One-Lead Bi-Gram of Normalized PSSM (Feature vector size 400)

3.4.6 Torsional Angles Composition

Torsional angles composition is similar to PSSM composition and defined as below:

$$TA - C(j) = \frac{1}{L}\sum_{i=0}^{L} TA_{i,j}$$

These features are calculated for each columns of the respective matrix.

Algorithm 5: One-Lead Bi-Gram Feature Extraction

```
$N \leftarrow$ Matrix from which feature will be extracted;
$L \leftarrow$ Length of the Protein;
$C \leftarrow$ Number of matrix column;
$V \leftarrow$ Empty array of size $C \times C$;
for $k = 0;\ k < C;\ k = k + 1$ do
    for $l = 0;\ l < C;\ l = l + 1$ do
        $sum \leftarrow 0$;
        for $i = 0;\ i < L - 2;\ i = i + 1$ do
            $sum = sum + N_{i,k}N_{i+2,l}$;
        $V_{k,l} = \frac{sum}{L}$;
```

3.4.7 Auto Co-variances of Probabilities

Auto covariance of probability matrix is also calculated depending on a distance factor K_p. It is formally defined as below:

$$PM - AC(j,k) = \frac{1}{L}\sum_{i=0}^{L-k} PM_{i,j}PM_{i+k,j}$$

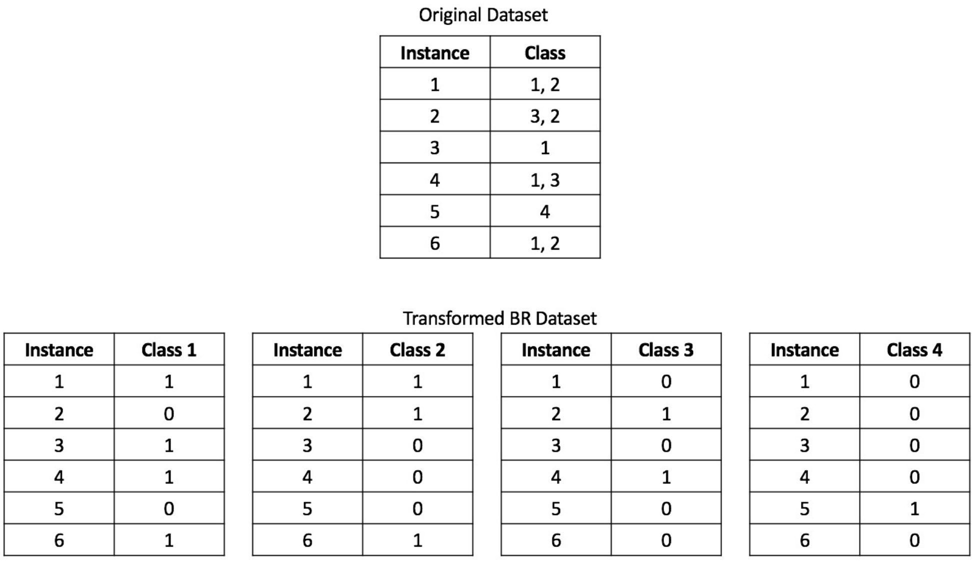
3.5 Multi-Label Learning

As we have used multi label data set, we analysis and compare with two encoding system. Our main methodologies goes with comparison of binary relevance with label power-set encoding.

3.5.1 Binary Relevance

Binary Relevance (BR) deal with dividing a multiple-choice problem into a series of yes/no questions, but the latter one evaluate with mutually exclusive choices. Problem transformation methods map the multi-label learning task into one or more single-label learning tasks. Multi-label problem is decomposed into several independent binary classification problems. Each label which participates in the multi-label problem.

The final multi-label prediction for a new instance is determined by aggregating the classification results from all independent binary classifiers. [2]



Original Dataset

Instance	Class
1	1, 2
2	3, 2
3	1
4	1, 3
5	4
6	1, 2

Transformed BR Dataset

Instance	Class 1
1	1
2	0
3	1
4	1
5	0
6	1

Instance	Class 2
1	1
2	1
3	0
4	0
5	0
6	1

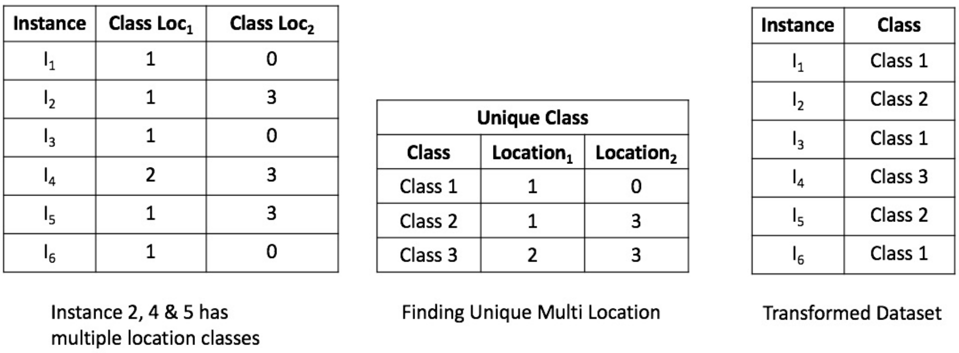
Instance	Class 3
1	0
2	1
3	0
4	1
5	0
6	0

Instance	Class 4
1	0
2	0
3	0
4	0
5	1
6	0

Figure 3.3: Binary Relevance (BR)

3.5.2 Label Power-set

The Label Power-set (LP) method removes the limitation of BR by taking into account label dependency. Label Power-set considers each unique occurrence of set of labels in multi label training dataset as one class for newly transformed dataset. Multi-label problem has been transformed into single multi-class unique-label learning problem. Where they used as target values for the class attribute all unique subsets of multi-labels [2].



Instance	Class Loc$_1$	Class Loc$_2$
I_1	1	0
I_2	1	3
I_3	1	0
I_4	2	3
I_5	1	3
I_6	1	0

Instance 2, 4 & 5 has multiple location classes

Unique Class		
Class	Location$_1$	Location$_2$
Class 1	1	0
Class 2	1	3
Class 3	2	3

Finding Unique Multi Location

Instance	Class
I_1	Class 1
I_2	Class 2
I_3	Class 1
I_4	Class 3
I_5	Class 2
I_6	Class 1

Transformed Dataset

Figure 3.4: The Label Power-set (LP)

Label power set relationships can be extracted from the training instances. In the LP method, for example, inter-relationships among labels are feed directly from the data. Multi-label methods are capable of handling the different relationships between labels. Like label dependency, cooccurrence and correlation. As existing combinations of single-labels present in the training instances, They can be used as a possible label in the correspondent multi-class in single-label classifier.

We use the label transformation method using label powerset encoding. Apart from label powerset, other successful method which is traditionally used in protein subcellular localization is binary relevance method. Let us suppose that for the multi-label classification problem, the set of different labels is $\mathcal{L} = \{1, 2, 3, \cdots, K\}$. We define a label, $\mathcal{Y} \subseteq \mathcal{L}$. Now the training set for any multi-label classification becomes, a set $\mathbb{S} = \{(\vec{x}_i, \mathcal{Y}_i) : \vec{x}_i \in \mathbb{R}^n, 1 \leq i \leq m\}$. Here, m is the number of total training instances or individual unique proteins and n is the dimensionality of the feature space. In binary relevance method, multiple learners are trained using the training dataset $\mathbb{S}$, each for labels, $l \in \mathcal{L}$. Predictions from each of these binary classifiers are then merged together in a vector $\{\hat{y}^1(\vec{x}, \hat{y}^2(\vec{x}, \cdots, \hat{y}^K(\vec{x})\}$. Elements of this vector are either 0 or 1 and the predicted labels are decided from this vector. However, in case of label powerset encoding preprocessing is required on the label space. The pseudo-code of the label powerset encoding is given in Algorithm 6.

```
**Algorithm 6:** Label Power-set
pre-process:
let $B$ a bijective function
$\mathbb{P} = \{1, 2, 3, \cdots, 2^K\}$ learn $B$ by mapping each label $\mathcal{Y}_i \in \mathbb{S}$ to a
    hyper-label $y_i \in \mathbb{P}$
$\mathbb{T} = \mathsf{transform}(\mathbb{S}, B)$
train:
learn a single multi-class classifier $h(\vec{x})$ from $\mathbb{T}$
predict:
for each $\vec{x}$, return $B^{-1}(h(\vec{x}))$
```

Label power-set encoding transforms each multi-label of single labels to a hyper label. This transformation maps each previous label to the power super set of the possible labels. A single base classifier is learned for the transformed training dataset and while prediction after the labels are decided by the classifier they are transformed back to the original labels. Note that compared to the binary relevance, here only a single learner has to be trained and hence reduces training time of the dataset.

To present the bijective mapping, lets illustrate the idea with an example. In the first phase new hyper labels are determined by taking combinations of labels. Note that all combinations are not present in the original dataset as shown in Figure 3.5(a). We could easily see that only three unique combinations are present in this dataset. The bijective mapping learns the new labels and transforms the whole dataset into a new one as shown in Figure 3.5(b).

Instance	$Location_1$	$Location_2$
I_1	1	0
I_2	1	3
I_3	1	0
I_4	2	3
I_5	1	3
I_6	1	0

(a)

Instance	Class
I_1	class 1
I_2	class 2
I_3	class 1
I_4	class 3
I_5	class 2
I_6	class 1

(b)

Figure 3.5: Illustration of Label power-set encoding, (a) original dataset and (b) transformed dataset.

Chapter 4

Experimentation, Result and Discussion

4.1 Feature Extraction

In our research first step is to extract desired features from desired data sets. In **Chapter 3** a brief description has been given about dataset, manipulation of dataset and feature extraction methods. We have used a dataset named as Gram-Negative Bacteria Protein dataset. We have implemented total 7 types (Composition, Obe-Lead Bi-Gram, Auto Covariance, PSSM-AAO, PSSM-SD, PSSM-SAC, PSSM-AC) of feature extraction methods from where we have extracted total 46 features per data set.

4.2 Choosing Classifier and Parameter Tunning

For choosing appropriate classifiers firstly we have extracted those features which have been reported in these literature [10], [29]. These two literature have reported that they have used Support Vector Machine (SVM) with RBF kernel and used 3000 as a value for cost parameter (C), 0.005 as a value for gamma (γ). We have tried other classifier such as Naive Bayes, Nearest Neighbor, Decision Tree and Random Forest, but these classifiers did not gave the promising result. We have also tried a little to change SVM kernal and tune parameters (C & γ). But we did not get the promising result. So for our experimental purpose we have chosen Support Vector Machine (SVM) as our classifier and values for γ and C are 0.005 and 3000. Using this classifier and parameters we have run our experiment to get final features set among 46 features. After getting the final features set we have tried to tune SVM parameters, but these did not give us the good result. So in this paper our reported classifier is Support Vector Machine (SVM), kernal is RBF, value of C is 3000 and value of γ is 0.005.

4.3 Choosing Validation Method

There are two types of validation methods used in this paper, one is 10-fold cross validation and another one is jackknife test also named as leave-one-out cross validation.

10-Fold Cross Validation: The original sample in 10-fold cross-validation, is randomly segregated into 10 equal portioned subsamples. Of those 10, one subsample is reserved for the validation in the testing model. The other 9 subsamples are considered to be training data. This mentioned cross-validation process is repeated 10 times. In this process each of the 10 subsamples are used only once as validation data. Now we have got 10 results from the folds then, we calculate the averaged to get a single estimation. The good thing about of this method is all observations are used throughly, by thoroughly means each observation is used for validation exactly once for both training and validation.

Jackknife Test: Jackknife Test, though it sounds sadistic but this is an interesting test. To describe the test we assume that there is N number of samples in the dataset. To run the test given sample is randomly divided into N equal sized subsamples. Then again, One subsamples is reserved as validation data for the testing model and remaining N-1 subsamples as considered as training data. Therefore this test also known as leave one out cross validation method.

While we run the test we repeat the process N times here, each of the N subsamples will be processed exactly once as validation data. To get a single estimation we will have to average those N results from the folds. Benefit of Jack-Knife test is similar to 10-Fold cross validation. But here is an additional disadvantages in this compare to other tests, it takes more time to complete the full process.

For our primary experiment we have performed 10-fold cross validation and for our final experiment we have performed jackknife test. Our main target for this experiment is to find out the novel features among our selected 46 features that gives the best result. As there are lots of combination (about $2^{46} - 1$ combinations, though we have not performed all this combinations, but we have performed a lots of combinations by adopting some mechanism which will be discussed in later section which reduces the combination size) for feature and 10-fold cross validation takes less time and is a remarkable validation method, therefor we have performed 10-fold cross validation for selecting best features. After selecting best features we have performed jackknife test as it has been widely used in the literature for this task and has been shown to be the most consistent and reliable method. In this thesis paper all our reported result is using jackknife test.

4.4 Sensitivity, Specificity and MCC

To provide more information about the statistical significance of our achieved results, we have also performed Sensitivity, Specificity and Matthew's Correlation Coefficient (MCC). Sensitivity, specificity and MCC are statistical measures of the performance of a binary classification test, also known in statistics as classification function. Sensitivity (also called the true positive rate, the recall, or probability of detection in some fields) measures the proportion of positives that are correctly identified. Specificity (also called the true negative rate) measures the proportion of negatives that are correctly identified. The value of sensitivity and specificity varies between 0 and 1. Having specificity, and sensitivity equal to 1 represents a fully accurate model while 0 represents a fully inaccurate. On the other hand, MCC measures the prediction quality of the model. MCC takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC is in essence a correlation coefficient between the observed and predicted binary classifications; it returns a value between 1 and +1. A coefficient of +1 represents a perfect prediction, 0 no better than random prediction and 1 indicates total disagreement between prediction and observation. The equation for calculating sensitivity, specificity and MCC are given below:

$$Sensitivity = \frac{TP}{TP + FN} \times 100$$

$$Specificity = \frac{TN}{TN + FP} \times 100$$

$$MCC = \frac{(TN \times TP) - (TN \times FP)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \times 100$$

where TP (true positive) is the number of correctly identified samples, FN (false negative) is the number of incorrectly rejected samples, TN (true negative) is the number of correctly rejected samples, and FP (false positive) is the number of incorrectly accepted samples.

4.5 Performance Evaluation

A number of sampling methods have been used in the literature for comparison of different prediction methods [18]. Among them are: percentage split, k-fold cross validation and jack knife tests. In most of the papers of protein subcellular localization, 10-fold cross validation of the learners have been widely applied [10,15,27,29] and also suggested in [4]. In this paper we also adopt 10-fold cross validation to validate our method and results with those of the state-of-the-art methods.

Accuracy for simple binary or multi-class classification problems are calculated by taking the percentage of true positives for each class to the total number of instances. However, as suggested in [5] such metrics could be mis-leading for classification of multi-label classification problems. Here, one should give importance to the accurate prediction of all multiple locations simultaneously. We define *absolute accuracy* for this purpose. Absolute accuracy was previously used in [29]. Absolute accuracy can be formally defined as below:

$$\text{absolute accuracy} = \frac{1}{N_{dif}} \sum_{i=1}^{N_{dif}} C_i \tag{4.1}$$

Here, N_{dif} is the total number of protein sequences in the dataset and $C_i = 1$ if all the locations of a protein is predicted correctly and otherwise, 0. Note that typical evaluations metrics like accuracy, sensitivity, specificity and others are only suitable for binary or multi-class classification problems and are not recommended for multi-label classification problem. [5].

4.6 Effect of Using Label Power-set Encoding

The first set set of experiments were done on the gram negative bacteria protein to show the effectiveness of the label power-set encoding with that of binary relevance. We tested our method with three base classifiers: SVM, decision tree J48 algorithm and Random Forest algorithm. For each of these classifiers, we used same set of hyper parameters and run each of them 5 times for binary relevance and label power-set encoding. Average and maximum absolute accuracy and standard deviation for each of the classifiers using two different schemes are given in Table 4.1.

		J48	SVM	RF
Binary Relevance	Avg	42.14	71.74	55.16
	Max	45.71	78.82	62.16
	St. Dev	3.49	3.49	3.49
Label Powerset	Avg	**60.84**	**82.09**	**69.97**
	Max	**64.87**	**83.65**	**72.21**
	St. Dev	3.18	1.14	2.23

Table 4.1: Performance Comparison of binary relevance and label power-set encoding on gram negative bacteria.

Our second set set of experiments were done on the pant protein dataset to show the effectiveness of the label power-set encoding with that of binary relevance. are given in Table 4.2.Best values in Tables 4.1 and 4.2 are shown in bold faced fonts. From the reported results it is clear that for all three classifiers label power-set method is achieving much higher absolute accuracy in comparison to the binary relevance method.

		J48	SVM	RF
Binary Relevance	Avg	31.58	19.76	28.74
	Max	34.02	21.23	3.052
	St. Dev	0.56	0.33	0.45
Label Powerset	Avg	**33.86**	**20.68**	**29.38**
	Max	**41.23**	**32.70**	**35.54**
	St. Dev	1.69	2.75	7.41

Table 4.2: Performance Comparison of binary relevance and label power-set encoding on Plant protein dataset.

The trend is similar for both in terms of maximum accuracy and average accuracy. We also plot the bars in Figure 4.1 to show a clear comparison. We also note that the standard deviation in the results are somehow lower in case of label power-set encoding. Also note that among all the classifiers Support Vector Machines (SVM) with rbf kernel performed the best and achieved superior performance. Thus we select SVM as a classifier for our proposed method.

4.7 Comparison with Other Methods

Method Name	Reference	Absolute Accuracy
Pacharawongsakda et al.	[25]	73.2%
Dehzangi et al.	[10]	76.6%
Dehzangi et al.	[16]	79.6%
Our method	This paper	**83.65%**

Table 4.3: Comparison of the absolute accuracy of our method with other state-of-the-art methods.

We also compare the performance of our method to that of other predictors in the literature. We compare our results with three other methods in the literature that predict gram negative bacterial proteins subcellular localization.

These methods are from Pacharawongsakda et al. [25], Dehzangi et al. [10] and Dehzangi et al. [16]. These all three methods use binary relevance on the protein labels and 10-fold cross validation. The reported

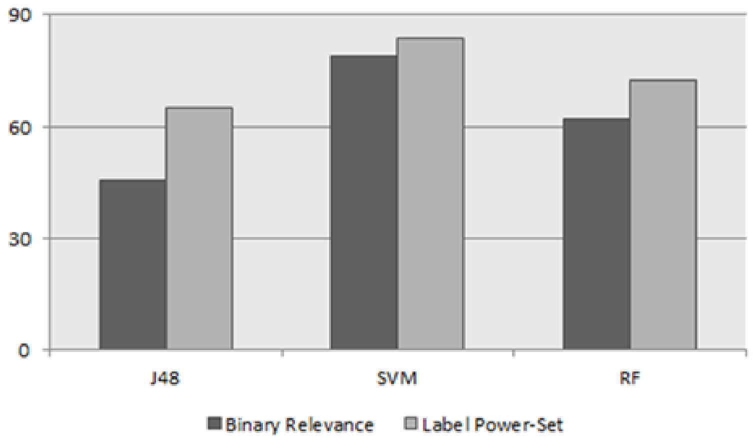


Figure 4.1: Bar plot showing the absolute accuracy achieved by different classifiers using label power-set encoding compared to binary relevance.

accuracy for these methods are taken from their papers and shown in Table 4.3. It is clear from the results shown in Table 4.3 that our method is able to produce superior results compared to these state-of-the-art predictors.

One of the potential drawback of our method is the hyper labeling. It solely depends on the success of this hyper labeling. Now, if any combinations of protein locations are missing in the training data but are present in the test data might hamper the performance of the label power-set encoding. We believe our method is more efficient and effective in datasets like gram negative and others where the number of different locations are higher in number. It is often noticed that the combinations of locations follow a specific pattern. However, in comparison of training time label power-set seems to be a definite winner.

Chapter 5

Conclusion

5.1 Summary

In our study we are trying to propose that implementing label powerset analysis for multi location protein classes will give us better result over binary relevance. Still we are trying to improve our output result by improving the feature extraction and trying to out more effective classifier for the multi level data sets.

Subcellular localization of Multi-class protein is a very important problem to solve in cell biology. One of the major challenge to solve this problem is the nature of the proteins that makes them located in multiple locations simultaneously. Thus the problems belongs to the category of multi-label classification. Traditional approaches in prediction of Multi-class protein locations binary relevance is used. In this paper, we proposed to use label space transformation using label power-set encoding. Our method was able to produce significantly improved results on a standard benchmark dataset and also tested of a number of classifiers shown promising results. We further wish to test our method on other multi-label datasets like plant or human datasets. We also wish to develop a web based tool for the biologists so that that can use it for practical purposes. We also believe that suitable feature selection technique and other features could result in enhancement of the proposed method.

5.2 Limitations

Though we have done a lot of experiments for producing the best result, there are some limitations in our work. These limitations are described below:

1. We have used label power set encoding in a fewer number of multi class protein dataset.

2. We can use more multi-class dataset to evaluate our methods more accurately.

3. In this experiment we have only tried one types of protein dataset (gram-negative bacteria protein and plant protein dataset).

4. We have only extracted PSSM-SD feature from PSSM matrix not from Spider SPD3 matrix.

5. In PSSM-SD feature extraction method we have only tried three values (5, 10 and 25) for F_p.

6. We have only extracted PSSM-SAC feature from PSSM matrix not from Spider SPD3 matrix.

7. In PSSM-SAC feature extraction method we have only tried ten values (1,2,3,.......,8,9,10) for K_p.

8. In Auto Covariance feature extraction method we have only tested 15 values (1,2,3,........13,14,15) for distance factor (DF) and took $DF = 10$ as it is reported in literature [29] which gives the best result and we have also investigated that this gives the best result among these 15 values.

9. In this literature we have only used Support Vector Machine (SVM) as our classifier.

10. In this experiment we have performed a little optimization for SVM parameters (γ and C).

5.3 Future Work

As there are some limitations in our current work, so we have planned to eliminate these limitations in future work. Besides this we have planned to do some extra things which may increase our accuracy rate. These future works are described below:

1. Try with more Multi class protein dataset.

2. Try new different feature extraction methods.

3. We will try to extract PSSM-SD feature from Spider matrix.

4. In PSSM-SD feature extraction method we will try other value of F_p along with previous value to extract features from both PSSM matrix and Spider matrix.

5. We will try to extract PSSM-SAC feature from Spider matrix.

6. In PSSM-SAC feature extraction method we will try other value of K_p along with previous value to extract features from both PSSM matrix and Spider matrix.

7. In Auto Covariance feature extraction method we will try other valu es along with previous values which have been mentioned in previous section for distance factor (DF) to see the effectiveness of our result.

8. We will try other supervised learning algorithm for e.g. Decision Tree, Random Forest, Naive Base Classifier, K-Nearest Neighbor, AdaBoost etc along with SVM to see the effectiveness of our result.

9. We will try to optimize SVM parameters (γ and C) more to see whether it increases the accuracy or not.

10. We will try to make a web application for global users, so that they can give a protein or amino acids sequence as input and our web application will analyze it and predict which location(s) this protein sits as an output. This is our main goal in future.

11. We will try our new method and technique to predict other protein dataset such as eukaryotic, human, animal, virus protein dataset etc.

References

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

[2] Everton Alvares Cherman, Maria Carolina Monard, and Jean Metz. Multi-label problem transformation methods: a case study. *CLEI Electronic Journal*, 14(1):4–4, 2011.

[3] Kuo-Chen Chou. Prediction of protein structural classes and subcellular locations. *Current protein and peptide science*, 1(2):171–208, 2000.

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

[5] Kuo-Chen Chou. Some remarks on predicting multi-label attributes in molecular biosystems. *Molecular Biosystems*, 9(6):1092–1100, 2013.

[6] Kuo-Chen Chou and Hong-Bin Shen. Large-scale predictions of gram-negative bacterial protein subcellular locations. *Journal of proteome research*, 5(12):3420–3428, 2006.

[7] Kuo-Chen Chou and Hong-Bin Shen. Cell-ploc: a package of web servers for predicting subcellular localization of proteins in various organisms. *Nature protocols*, 3(2):153–162, 2008.

[8] Kuo-Chen Chou, Hong-Bin Shen, et al. Cell-ploc 2.0: An improved package of web-servers for predicting subcellular localization of proteins in various organisms. *Natural Science*, 2(10):1090, 2010.

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

[10] Abdollah Dehzangi, Rhys Heffernan, Alok Sharma, James Lyons, Kuldip Paliwal, and Abdul Sattar. 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, 2015.

[11] Abdollah Dehzangi, Kuldip Paliwal, Alok Sharma, Omid Dehzangi, and Abdul Sattar. 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, 2013.

[12] Abdollah Dehzangi, Kuldip K Paliwal, Alok Sharma, James G Lyons, and Abdul Sattar. Protein fold recognition using an overlapping segmentation approach and a mixture of feature extraction models. In *Australasian Conference on Artificial Intelligence*, pages 32–43. Springer, 2013.

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

[14] Abdollah Dehzangi and Abdul Sattar. Ensemble of diversely trained support vector machines for protein fold recognition. In *ACIIDS (1)*, pages 335–344, 2013.

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

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

[17] Qiwen Dong, Shuigeng Zhou, and Jihong Guan. A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. *Bioinformatics*, 25(20):2655–2662, 2009.

[18] Bradley Efron and Gail Gong. A leisurely look at the bootstrap, the jackknife, and cross-validation. *The American Statistician*, 37(1):36–48, 1983.

[19] Maryam Esmaeili, Hassan Mohabatkar, and Sasan Mohsenzadeh. Using the concept of chou's pseudo amino acid composition for risk type prediction of human papillomaviruses. *Journal of theoretical biology*, 263(2):203–209, 2010.

[20] Yanzhi Guo, Menglong Li, Minchun Lu, Zhining Wen, and Zhongtian Huang. Predicting g-protein coupled receptors–g-protein coupling specificity based on autocross-covariance transform. *Proteins: structure, function, and bioinformatics*, 65(1):55–60, 2006.

[21] Yanzhi Guo, Lezheng Yu, Zhining Wen, and Menglong Li. Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. *Nucleic acids research*, 36(9):3025–3030, 2008.

[22] Rhys Heffernan, Yuedong Yang, Kuldip Paliwal, and Yaoqi Zhou. Capturing non-local interactions by long short term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers, and solvent accessibility. *Bioinformatics*, page btx218, 2017.

[23] Chao Huang and Jingqi Yuan. Using radial basis function on the general form of chou's pseudo amino acid composition and pssm to predict subcellular locations of proteins with both single and multiple sites. *Biosystems*, 113(1):50–57, 2013.

[24] Taigang Liu, Xiaoqi Zheng, Chunhua Wang, and Jun Wang. Prediction of subcellular location of apoptosis proteins using pseudo amino acid composition: an approach from auto covariance transformation. *Protein and peptide letters*, 17(10):1263–1269, 2010.

[25] Eakasit Pacharawongsakda and Thanaruk Theeramunkong. Predict subcellular locations of singleplex and multiplex proteins by semi-supervised learning and dimension-reducing general mode of chou's pseaac. *IEEE transactions on nanobioscience*, 12(4):311–320, 2013.

[26] Kuldip K Paliwal, Alok Sharma, James Lyons, and Abdollah Dehzangi. 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, 2014.

[27] Harsh Saini, Gaurav Raicar, Abdollah Dehzangi, Sunil Lal, and Alok Sharma. Subcellular localization for gram positive and gram negative bacterial proteins using linear interpolation smoothing model. *Journal of theoretical biology*, 386:25–33, 2015.

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

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

[30] Hong-Bin Shen and Kuo-Chen Chou. Gpos-ploc: an ensemble classifier for predicting subcellular localization of gram-positive bacterial proteins. *Protein Engineering Design and Selection*, 20(1):39–46, 2007.

[31] Shibiao Wan, Man-Wai Mak, and Sun-Yuan Kung. Goasvm: a subcellular location predictor by incorporating term-frequency gene ontology into the general form of chou's pseudo-amino acid composition. *Journal of Theoretical Biology*, 323:40–48, 2013.

[32] S Wold, J Jonsson, M Sjörström, M Sandberg, and S Rännar. Dna and peptide sequences and chemical processes multivariately modelled by principal component analysis and partial least-squares projections to latent structures. *Analytica Chimica Acta*, 277(2):239–253, 1993.

[33] Jiang Wu, Meng-Long Li, Le-Zheng Yu, and Chao Wang. An ensemble classifier of support vector machines used to predict protein structural classes by fusing auto covariance and pseudo-amino acid composition. *The protein journal*, 29(1):62–67, 2010.

[34] Li Yang, Yizhou Li, Rongquan Xiao, Yuhong Zeng, Jiamin Xiao, Fuyuan Tan, and Menglong Li. Using auto covariance method for functional discrimination of membrane proteins based on evolution information. *Amino Acids*, 38(5):1497–1503, 2010.

[35] Yu-hong Zeng, Yan-zhi Guo, Rong-quan Xiao, Li Yang, Le-zheng Yu, and Meng-long Li. Using the augmented chou's pseudo amino acid composition for predicting protein submitochondria locations based on auto covariance approach. *Journal of theoretical biology*, 259(2):366–372, 2009.