Heaven's Light is Our Guide

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Rajshahi University of Engineering & Technology, Bangladesh

Prediction of Lysine Glycation PTM site in Protein using Peptide Sequence Evolution based Features

Author
S.M.Shovan
133001
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology

Supervised by
Dr. Md. Al Mehedi Hasan
Associate Professor
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology

ACKNOWLEDGEMENT

First of all, all praises and thanks to the Almighty Allah for bestowing His blessing upon me to complete this task.

Then I would like to express my heartfelt gratitude and respect to my honorable supervisor Dr. Md. Al Mehedi Hasan, Associate Professor, Department of Computer Science & Engineering, Rajshahi University of Engineering & Technology, Rajshahi.,for his careful guidance, sincere advice, consistent support, adequate encouragement and ever possible help throughout my entire candidature.

I want to express gratitude and thanks to Professor Dr. Md. Rabiul Islam, Head, Department of Computer Science & Engineering, Rajshahi University of Engineering & Technology, for his extending helps in various ways from his department.

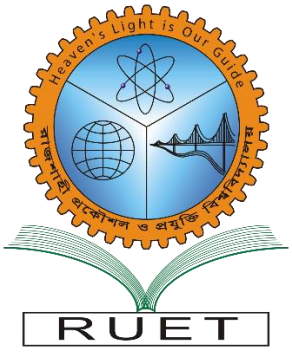
I am grateful to others teachers of Department of Computer Science & Engineering, Rajshahi University of Engineering & Technology, Rajshahi, for their kind suggestions and encouragements from time to time.

Finally, I would like to thank my parents, sister, brother, friends and well-wishers for their continuous inspiration and endeavor throughout the work.

12.11.2018
RUET, Rajshahi

S.M.Shovan

Heaven's Light is Our Guide



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Rajshahi University of Engineering & Technology, Bangladesh

CERTIFICATE

This is to certify that this thesis report entitled "**Prediction of Lysine Glycation PTM site in Protein using Peptide Sequence Evolution based Features"** submitted by **Name: S.M. Shovan, Roll: 133001** in partial fulfillment of the requirement for the award of the degree of Bachelor of Science in Computer Science & Engineering of Rajshahi University of Engineering & Technology, Bangladesh is a record of the candidate own work carried out by him under my supervision. This thesis has not been submitted for the award of any other degree

Supervisor

Dr. Md. Al Mehedi Hasan

Associate Professor

Department of Computer Science &Engineering

Rajshahi University of Engineering &Technology

Rajshahi-6204

External Examiner

Department of Computer Science &Engineering

Rajshahi University of Engineering &Technology

Rajshahi-6204

12.11.2018
RUET, Rajshahi

ABSTRACT

Glycation is a post-transactional modification which is non-enzymatic in nature. It is closely associated with different biological functions and responsible for many diseases, for example, diabetes, renal failure etc. Identification of Glycation sites is very important in the development of drugs and the research areas but identifying manually in laboratory is laborious, costly and time consuming. Development of a computational tool will be very useful for the prediction of Glycation sites with a high accuracy. In our experiment, a new feature extraction technique, called peptide sequence evolution based feature representation, is introduced which gave an Accuracy of 95.94±0.54%, a Sensitivity of 98.20% and a Specificity of 90.67±1.07% after running 10-fold cross-validation five times. This result outperforms the previously developed tools BPB_GlySite, NetGlycate, PreGly and Gly_PseAAC.

CONTENTS

CHAPTER 1

Introduction

CHATER 2

Post-Translational Modification

CHAPTER 3

Data Preprocessing

CHAPTER 4

Feature Extraction

CHAPTER 5

Support Vector Machine Classifier

CHAPTER 6

Measuring Matrices

CHAPTER 7

Implementation and performance analysis

CHAPTER 8

Conclusion

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Overview

Proteins consist of one or more long chains of amino acid residues. It performs a vast array of functions within organisms, including catalyzing metabolic reactions, DNA replication, responding to stimuli, providing structure to cells and organisms, and transporting molecules from one location to another. Central dogma is an explanation of the flow genetic information within a biological system. It explains how proteins are formed from the DNA. Protein goes through several modifications after the biosynthesis of the proteins. These modifications refers to Post-Translational modification (PTM) which is a covalent and generally enzymatic modification. Normally these modifications occur in the side chain of amino acids. These PTMs are responsible for the diversity of protein in terms of their functions and structures as well as the plasticity and dynamics of active cells are greatly influenced by PTMs [1]. Expansion of genetic code and regulation of physiology of the living cells are the results of PTMs [2,3]. Moreover, PTMs can also cause several human diseases. So, identification of PTMs are necessary in the drug development and also in the research areas.

1.1 Concepts of Protein

Proteins are large biomolecules, consisting of one or more long chains of amino acid residues. Amino acids are organic compounds containing amine (-NH2) and carboxyl (-COOH) functional groups, along with a side chain (R group) specific to each amino acid. The key elements of an amino acid are carbon I, hydrogen (H), oxygen (O), and nitrogen (N) [4].

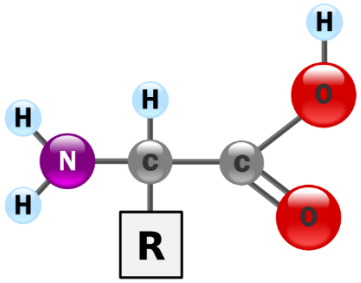


Figure 1.1 Structure of amino acids [4]

There are 20 different amino acids those form proteins. R group is the side chain of amino acids. Normally, the modification occurs in the side chain of amino acids which leads to the diversity and plasticity of proteins in terms of their structures and functionalities. A residue in biochemistry and molecular biology refers to a specific monomer, a single unit molecule, within the protein or nucleic acid. Peptides are the short chain of amino acid monomers or residues which are linked with peptide bonds. Sometimes, protein containing less than 20-30 residues are called peptides or oligopeptides and a chain of long amino acid residues is called polypeptide. Generally, this polypeptide is considered as proteins.

1.2 Formation of protein

The formation of protein from DNA is called central dogma, which explains the flow of genetic information within a biological system [5]. The central dogma consists of three steps.

1. **DNA replications:** Information is copied from DNA to DNA is the fundamental step in the central dogma.
2. **Transcription:** Transcription is the process by which the information contained in a section of DNA is replicated in the form of a newly assembled piece of messenger RNA (mRNA). Enzymes facilitating the process include RNA polymerase and transcription factors.
3. **Translation:** Proteins is synthesized using the information in mRNA (messenger RNA) as a template while amino acids are carried out by tRNA (transfer RNA).

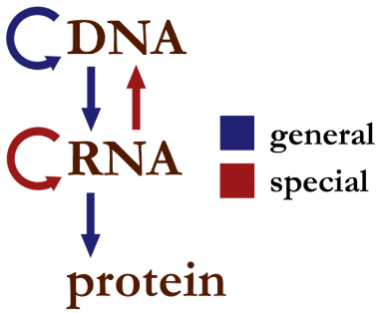


Figure 1.2 Information flow in biological systems [6]

1.3 Representation of protein

As we already know that proteins are long chain of amino acids. There are 20 different amino acids. Each proteins are the combination of 20 different amino acids with arbitrary length. The characteristics of proteins differs from the pattern and combination of amino acids. To represent amino acids, there two different notations. The following table consists of 3-Letter and 1-Letter amino acid notations [7].

Table 1.1 Twenty natural amino acids with notations

Amino Acid	3-letter [7]	1-Letter [7]
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartic acid	Asp	D
Cysteine	Cys	C
Glutamic acid	Glu	E
Glutamine	Gln	Q
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanine	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

1.4 Post-translational modification

PTM stands for post-translational modification. The enzymatic or non-enzymatic modification of protein after the biosynthesis is called post-translational modification. In later PTM will be discussed extensively. Normally PTMs are responsible for the diversity of protein in terms of their functions and structures as well as the plasticity and dynamics of active cells are greatly influenced by PTMs [1]. Expansion of genetic code and regulation of physiology of the living cells are the results of PTMs [2,3]. Moreover, PTMs can also cause several human diseases. Lysine glycation modification can cause several human disease, such as, vascular complications for diabetes [8], Parkinson's disease and Alzheimer's disease [9], renal failure [10]. In the later chapters, PTM will be discussed extensively.

1.5 Motivations

As we already know, the proteins suffer from several types of modifications. These modifications occur at the side chain of amino acids. For each modifications, there is a specific set of residues on which the modifications can occur, called as responsible residue. For example, glycation is the modification that can occur only on lysine. Though these modification is responsible for the diversities of the functionality of protein, it also results in several diseases. For instance, glycation modification can cause several human disease, such as, vascular complications for diabetes [8], Parkinson's disease and Alzheimer's disease [9], renal failure [10]. So it will be very useful to identify the modifications for the treatment of different diseases as well as in the drug development.

1.6 Objectives

The main objective of the experiment is to develop a computational tool which will effectively predict the protein whether they are glycated or not without the help of laboratory experiment. The main reason behind this is, the laboratory method is expensive, requires human effort and interaction, time consuming as well as laborious. So, if we can find a tool that will predict the protein whether they are glycated or not, it will both save our time and money.

1.7 Related works

This article consists of those paper which are related with this study. These papers are mostly selected on the basis of previous works that has been done on the prediction of lysine glycation sites. Three things of these paper has been shown and considered most important in our analysis. These are the PTM on which the paper is written on, feature extraction techniques and the classification techniques. The papers are listed below.

- M. Johansen, L. Kiemer and S. Brunak, "Analysis and prediction of mammalian protein glycation", Glycobiology, vol. 16, no. 9, pp. 844-853, 2006 [11].
  - PTM: Glycation
  - Predictor is built combining 60 artificial neural networks in a balloting procedure

- Y. Liu, W. Gu, W. Zhang and J. Wang, "Predict and Analyze Protein Glycation Sites with the mRMR and IFS Methods", BioMed Research International, vol. 2015, pp. 1-6, 2015 [12].
  - PTM : Glycation
  - Feature extraction: Composition of k-spaced amino acid pairs
  - Classifier: Support vector machine

- M. Hasan, J. Li, S. Ahmad and M. Molla, "predCar-site: Carbonylation sites prediction in proteins using support vector machine with resolving data imbalanced issue", Analytical Biochemistry, vol. 525, pp. 107-113, 2017 [13].
  - PTM: Carbonylation
  - Feature extraction: Sequence coupling model
  - Classifier: Support vector machine

- M. Hasan, S. Ahmad and M. Molla, "iMulti-HumPhos: a multi-label classifier for identifying human phosphorylated proteins using multiple kernel learning based support vector machines", Molecular BioSystems, vol. 13, no. 8, pp. 1608-1618, 2017.

- PTM : Phosphorylation
- Feature extraction: Multi-kernel
- Classifier: Support vector machine

- Y. Xu, L. Li, J. Ding, L. Wu, G. Mai and F. Zhou, "Gly-PseAAC: Identifying protein lysine glycation through sequences", Gene, vol. 602, pp. 1-7, 2017.
  - PTM: Glycation
  - Feature extraction: Position-specific amino acid propensity (PSAAP)
  - Classifier: Support vector machine

- Z. Ju, J. Sun, Y. Li and L. Wang, "Predicting lysine glycation sites using bi-profile bayes feature extraction", Computational Biology and Chemistry, vol. 71, pp. 98-103, 2017.
  - PTM: Glycation
  - Feature extraction: Bi-Profile Bayes (BPB)
  - Classifier: Support vector machine

1.8 Outline of the paper

The thesis paper contains 7 chapters in total. Here is the brief discussions of each chapters.

- **Chapter 1**

  Contains the brief discussion about the protein and the biosynthesis process of proteins as well as shows how the amino acids can form different proteins. The short discussion of amino acids includes the names, 3-letter and 1 letter notations and their significance on protein formation. This chapter also focuses on the objectives and motivations of this book.

- **Chapter 2**

  Chapter 2 is all about post-translational modifications. This chapter includes the introduction of several modifications with respect to the responsible residues, how the modification occurs is visualized with figures, effects and the consequences of

these modifications. This chapter is necessary to have a basic idea of different PTMs and their workings.

- **Chapter 3**

  Chapter 3 is all about data preprocessing. Data preprocessing is a vast topic in the data mining. But this chapter only covers the issues those are related to the topic of our studies, for example, how peptide sequences are generated from protein sequences while keeping the responsible residue in the middle. This chapter also includes several data balance technique which is necessary for handling the data imbalance issue which is persistent in most of the protein post-translational modification datasets.

- **Chapter 4**

  Feature extraction is one of the most important part of designing of predictor. The successfulness of predictor depends mostly on the feature extraction technique. In this chapter, several feature extraction technique has been discussed with mathematics as well as the technique proposed by us. The chapter only covers the techniques which have been implemented in the implementation chapter and keep other techniques out of the scope of this book.

- **Chapter 5**

  As a classifier, we have only used support vector machine for comparing the proposed model with the previously developed tools to understand how well our feature extraction technique performs against the existing techniques. This chapter contains mathematics and working principle as well as the parameters with the tuning method is covered.

- **Chapter 6**

  For measuring the performance of the predictor and to compare with the existing tools, sophisticated statistical measuring tools needs to be implemented. This

chapter covers most popular measuring matrices, confusion matrices, with sufficient equations and descriptions.

- **Chapter 7**

  The most important chapter in this book is the chapter 7, which includes the brief discussion about the dataset with samples. All the previous chapter has influence on this chapter. The system specification and parameters have shown in the tabular form and the results are show in both tabular and graphical representations for ensuring the intuitiveness of the understanding. This chapter reflects the outcome of our study.

Conclusion

In this chapter, we have discussed about the proteins and formation of the proteins briefly. This chapter also showed how amino acid acts as a building blocks for proteins. It covers the basic concepts of different amino acids and also some terminologies are discussed which will be necessary in the future chapters. This introductory chapter contains information those are necessary for getting the clear concept of our future discussions. It also clarifies the purposes, motivations and the objective of our thesis.

Chapter 2
Post-Translational Modification

Introduction

Post translational modification (PTM) is the modification of protein after the biosynthesis of protein. Translation is the last step of central dogma. As the modification occurs after the formation of protein, this is called the post-translational modification. Normally this modifications occur in the side chain of amino acids. Proteins go through many different types of modifications. This chapter will cover the basic concepts of post translational modification, types of PTMs and the results of it.

2.1 Concepts of PTM

PTM stands for post-translational modification, refers to the covalent modification of proteins following the protein biosynthesis. Proteins are synthesized by ribosomes translating mRNA into polypeptide chains, which may then undergo PTM to form the mature protein product. These modifications can be,

1. **Enzymatic:** Requires the presence of specific enzymes and enzymes plays an active role in the modifications. For instance, acylation, alkylation, glycosylation, hydroxylation etc.
2. **Non Enzymatic:** Does not require the presence of any enzymes. For instance, glycation, carbonylation etc.

Enzymes are macromolecular biological catalyst that influence that biological reactions in living entity. The enzymatic modification is more common in the PTMs. Some types of post-translational modification are consequences of oxidative stress. Overflow of free radical is called oxidative stress. Anti-oxidant helps by donating electron. Oxidative stress reflects an imbalance between the systemic manifestation of reactive oxygen species and a biological system's ability to readily detoxify the reactive intermediates or to repair the resulting damage. Carbonylation is one example that targets the modified protein for degradation and can result in the formation of protein aggregates. Specific amino acid modifications can be used as biomarkers indicating oxidative damage. A biomarker, or

biological marker, generally refers to a measurable indicator of some biological state or condition. The term is also occasionally used to refer to a substance whose detection indicates the presence of a living organism.

2.2 Modification process

Post-translational modifications can occur on the amino acid side chains or at the protein's C- or N- termini. The C-terminus (also known as the carboxyl-terminus, carboxy-terminus, C-terminal tail, C-terminal end, or COOH-terminus) is the end of an amino acid chain (protein or polypeptide), terminated by a free carboxyl group (-COOH). The N-terminus (also known as the amino-terminus, NH2-terminus, N-terminal end or amine-terminus) is the start of a protein or polypeptide referring to the free amine group (-NH2) located at the end of a polypeptide [17].

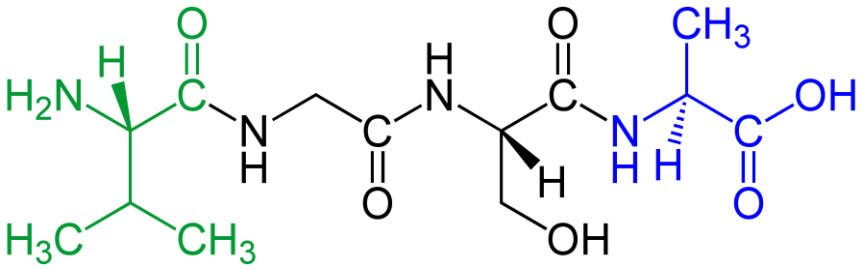


Figure 2.1 C- and N- termini of amino acids [18]

In the figure, blue portion represents the C- termini as it contains carboxyl group (-COOH) and green portion represents the N- termini as it contains amine group (-NH2).

Acetylation, a common PTM, is the process of introducing an acetyl function group into a chemical compound. N-Acetylation is an example of modification in the amino group. It is an enzymatic modification which is catalyzed by a set of enzyme complexes, the N-terminal acetyltransferases (NATs). NATs transfer an acetyl group from acetyl-coenzyme A (Ac-CoA) to the α-amino group of the first amino acid residue of the protein. Different NATs are responsible for the acetylation of nascent protein N-terminal, and the acetylation was found to be irreversible so far [19].

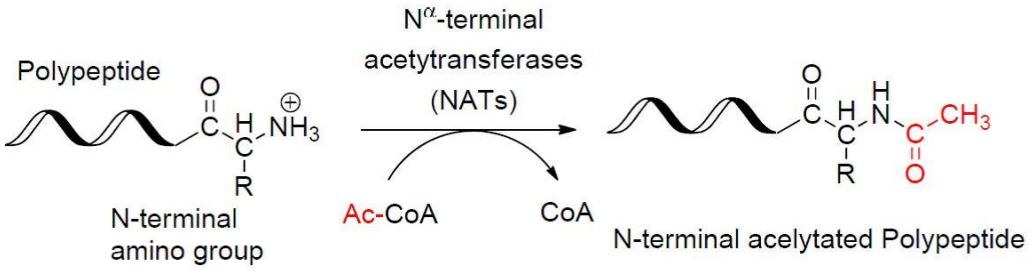


Figure 2.2 N-terminal acetylation [20]

The figure 2.2 show how N-terminal acetylation occurs with the presence of catalyst NATs.

2.3 Effects of PTMs

These PTMs are responsible for the diversity of protein in terms of their functions and structures as well as the plasticity and dynamics of active cells are greatly influenced by PTMs [1]. Expansion of genetic code and regulation of physiology of the living cells are the results of PTMs [2,3]. Sometimes, the post-translational modifications can results in several diseases. For example, glycation PTM can cause several human disease, such as, vascular complications for diabetes [8], Parkinson's disease and Alzheimer's disease [9], renal failure [10]. Carbonylation PTM can cause various types of major human diseases including Alzheimer's disease, diabetes, Parkinson's disease, chronic renal failure, chronic lung disease, sepsis are associated with protein carbonylation [21-25]. Phosphorylation shows a crucial role in the transmission of signals, which controls a diverse array of cellular functions, such as cell growth, survival differentiation, and metabolism [26].

2.4 Responsible residue of PTMs

Each post-translational modification has a set of residues on which that modification works on. For example, glycation modification only works on lysine (K) residue, called a responsible residue for glycation. A modification can occur on a set of residue. For example, responsible residue for carbonylation are lysine (k), proline (P), arginine I, threonine (T) [21-25]. The following table contains a set of post-translational modification with the respective residues.

Table 2.1 PTMs with responsible residues [27]

Amino Acid	Modifications
Alanine	N-acetylation (N-terminus)
Arginine	Methylation
Asparagine	N-linked glycosylation
Aspartic acid	Isomerization to isoaspartic acid
Cysteine	disulfide-bond formation, oxidation to sulfenic, sulfinic or sulfonic acid, palmitoylation, N-acetylation (N-terminus), S-nitrosylation
Glutamic acid	cyclization to Pyroglutamic acid (N-terminus), deamidation to Glutamic acid or isopeptide bond formation to a lysine by a transglutaminase
Glutamine	cyclization to Pyroglutamic acid (N-terminus), gamma-carboxylation
Glycine	N-Myristoylation (N-terminus), N-acetylation (N-terminus)
Histidine	Phosphorylation
Isoleucine	none
Leucine	none
Lysine	acetylation, Ubiquitination, SUMOylation, methylation, hydroxylation
Methionine	N-acetylation (N-terminus), oxidation to sulfoxide or sulfone
Phenylalanine	none
Proline	Hydroxylation
Serine	Phosphorylation, O-linked glycosylation, N-acetylation (N-terminus)
Threonine	Phosphorylation, O-linked glycosylation, N-acetylation (N-terminus)
Tryptophan	mono- or di-oxidation, formation of Kynurenine
Tyrosine	Sulfation, phosphorylation
Valine	N-acetylation (N-terminus)

In the table 2.1, All the modifications occur in the side chain of amino acid except those are explicitly specified as C-terminus or N-terminus.

Conclusion

In this chapter, we have discussed about what is post-translational modification and their effects. We also briefly showed about how the modification occurs on the side chain of protein or the C- or N- terminals. Different post-translational modifications have shown in the table 2.1 with the respective residues. The purpose of this chapter is clarify the concept of the modification that occurs in protein after the biosynthesis in brief. At the end of this chapter, we should have clear conception about different post-translational modifications.

Chapter 3
Data Preprocessing

Introduction

Data preprocessing is the first and foremost task before working with dataset. Data contains in the dataset can be missing, errors, inconsistent, duplicate or noisy. As we already know, protein is a long sequence of amino acids and each amino acid can be represented using 1-letter notation, then protein can be represented using a string of notations. As protein is a string, the character consisting the string can alter for any reason which can lead us to error as the domain of each 1-letter notation is 20, shown in table 1.1. In the following chapter, feature extraction has been discussed. For making suitable input from protein sequences, peptide sequence of a specific length is considered. The rules of generating peptide sequence form the protein sequence will be discussed in this section. In nature, data suffers from imbalance issue. For balancing out the imbalance several technique has been developed in the past years. In this chapter, we will learn what data imbalance is and the problems created by it as well as briefly go through several data balance techniques.

3.1 Generation of peptide sequence form protein sequence

Before feeding into feature extraction technique, subsets of the protein sequence is considered called the peptide sequence, form the total protein sequence. The process of considering the subset is shown below.

According to the iCar-PseCp [28], a peptide sample is represented as follows,

$$P_{\xi}(\odot) = R_{-\xi}R_{-(\xi-1)} \dots R_{-2}R_{-1}\Theta R_1 R_2 \dots R_{+(\xi-1)}R_{+\xi} \tag{3.1}$$

Here, Θ denotes responsible residue. Ξ is an integer. $R_{-\xi}$ represents upstream and $R_{+\xi}$ represents downstream of peptide sample. Total length of the peptide sample is $(2\xi+1)$.

Each peptide sample falls under one of the two categories as follows,

$$P_{\xi}(\odot) \in \begin{cases} P_{\xi}^{+}(\odot), & \text{if its center is a positive site} \\ P_{\xi}^{-}(\odot), & \text{otherwise} \end{cases} \tag{3.2}$$

$P_{\xi}^{+}(\odot)$ represents positive segments and $P_{\xi}^{-}(\odot)$ represents negative segments and $\in$ refers to membership of set theory.

Benchmark dataset has constructed as follows,

$$S_{\xi}(K) = S_{\xi}^{+}(K) \cup S_{\xi}^{-}(K), \text{ when } \Theta = K \quad (3.3)$$

Where, $S_{\xi}^{+}(\odot)$ contains positive segments, $P_{\xi}^{+}(\odot)$and $S_{\xi}^{+}(\odot)$ contains negative segments, $P_{\xi}^{-}(\odot)$ and $\cup$ is the union operation of set theory. Here positive means the modification has occurred and negative means modification not occurs in the peptide segment.

The basic idea is that,

- We need to identify the responsible residue for the specific post-translational modification.
- Find each residue from the protein and take the neighboring residues with the predefined length of upstream and downstream which will eventually create a peptide sequence of one plus two times the length of upstream and downstream. Here important factor is that the optimal length of upstream and downstream needs to be specified for optimal performance. As finding the optimal length is not a convex problem, we need to apply trial and error method for getting the best possible value for the length of upstream and downstream.
- If there are no enough residue either in the left or the right side of the responsible residue, we need to pad with a dummy reside “X” for ensuring the uniformity for each peptide sample.
- A class label is given 1, if the position of the responsible residue matches with the position of the occurrence of PTM given in the dataset. Otherwise, the class label is given 0. Here class label 1 means the positive site and class label 0 means the negative site.

Let's consider an example, A protein sequence is given by, MQMKATILIVLVALFMIQQSEAGWLRKAAKSVGKFYYKHKYYIKAAWQIGKHALGDMTDEEFQDFMKEVEQAREEELQSRQ.

Now consider K, lysine is the responsible residue, and the length of upstream and downstream is 7. So, the responsible residue K is marked as red in the following sequence, MQMKATILIVLVALFMIQQSEAGWLRKAAKSVGKFYYKHKYYIKAAWQIGKHALGDMTDEEFQDFMKEVEQAREEELQSRQ

The peptide sequences are,

1. XXXXMQMKATILIVL
2. SEAGWLRKAAKSVGK
3. GWLRKAAKSVGKFYY
4. KAAKSVGKFYYKHKY
5. SVGKFYYKHKYYIKA
6. GKFYYKHKYYIKAAW
7. YKHKYYIKAAWQIGK
8. KAAWQIGKHALGDMT
9. EEFQDFMKEVEQARE

There are 10 peptide sequence is found from the protein sequence using the technique discussed earlier in this section.

3.2 Data imbalance problem

A dataset is called imbalanced if it contains different number of data for each classes. For balancing out dataset, the number of data in each classes are made equal. There are several technique that helps for balancing dataset. The following section contains the brief description. Before understanding the techniques, first we need to understand what purpose of data balance is and what problems are created for the dataset with imbalanced data.

3.2.1 Identification of imbalanced dataset

Before we get started, it is necessary to identify whether the data is imbalanced and require to balance. First consider an example,

A utilities fraud detection data set have the following data:

- Total observations = 1000
- Fraudulent observations = 20
- Non fraudulent observations = 980
- Event Rate= 2 %

In this example, we have two classes, fraudulent and non-fraudulent. This is also called binary class for having exactly two classes. The fraudulent observation is said to be minority class as it contains less number of observations compared to non-fraudulent observations, so non-fraudulent class is said to be majority class. A dataset is said to be balanced if its imbalance ratio [29] is one according to the following formula.

$$\text{Imbalance ratio, IR} = \frac{\text{No of observation in majority class}}{\text{No of observation in minority class}} \tag{3.4}$$

$$= \frac{\text{No of observation in non}-\text{fradulant class}}{\text{No of observation in fradulant class}}$$

$$= \frac{980}{20}$$

$$= 49$$

Here, Imbalance ratio is 49, which is significantly large compared to 1, so we can say that the data is significantly imbalanced.

3.2.2 Effects of imbalanced dataset

Imbalanced dataset can bias the results for different performance measuring matrix of classifier. Let's consider a situation where a classifier always says non-fraudulent to example discussed in the article 3.3. Then for 1000 observation, the classifier accurately classifies 980 times and does error for only 20 times. So the accuracy will be, 98%. Though the predictor always says non-fraudulent, but the accuracy is high. In this situation, imbalanced dataset biased the accuracy. So we can draw a conclusion that, for the highly imbalanced data, accuracy is not a good measuring matrix for measuring the performance of the classifier.

Another issue regarding the data imbalance is that proper inspection is required while doing cross-validation for validating models. If the minority class is very low, observations of each example should contain in each fold, otherwise the performance of the classifier will be poor. After a brief discussion, it can be undoubtedly said that balancing the imbalance data is very important for different aspect.

3.2.3 Data balance techniques

Over years, several data balancing technique have been introduced, none are found to be flawless and perfect in all situations. All techniques have its own benefits as well as disadvantages. This section will cover several techniques.

Resampling technique is a category of technique that considers the frequency of each classes. The main objective of balancing classes is to either increasing the frequency of the minority class or decreasing the frequency of the majority class. This is done in order to obtain approximately the same number of instances for both the classes.

3.2.3.1 Random under-sampling

Random undersampling aims to balance class distribution by randomly eliminating majority class examples while keeping the minority class untouched. Here data is randomly selected without replacement to make equal probability for all the observation to be selected for elimination [29,30].

From the example in section 3.3, if we take 2% of data from the majority class, the IR, Imbalance ratio is close to 1.

- **Advantages**
    - It can help improve run time and storage problems by reducing the number of training data samples when the training data set is huge.

- **Disadvantages**
    - It can discard potentially useful information which could be important for building rule classifiers.

- The sample chosen by random under sampling may be a biased sample. And it will not be an accurate representative of the population. Thereby, resulting in inaccurate results with the actual test data set.

3.2.3.2 Random over-sampling

Over-sampling increases the number of instances in the minority class by randomly replicating them in order to present a higher representation of the minority class in the sample. Random selection is made with replacement for ensuring the uniformity in probability for each observation [29,31].

From the example in section 3.3, selecting observation randomly with replacement for n time such that, the IR, Imbalance ratio is close to 1.

- **Advantages**
  - Unlike under sampling this method leads to no information loss.
  - Outperforms under sampling
- **Disadvantages**
  - It increases the likelihood of overfitting since it replicates the minority class events.

3.2.3.3 Cluster based over-sampling

In this case, the K-means clustering algorithm is independently applied to minority and majority class instances [29.32]. This is to identify clusters in the dataset. Subsequently, each cluster is oversampled such that all clusters of the same class have an equal number of instances and all classes have the same size.

- **Advantages**
  - This clustering technique helps overcome the challenge between class imbalance. Where the number of examples representing positive class differs from the number of examples representing a negative class.
  - Also, overcome challenges within class imbalance, where a class is composed of different sub clusters. And each sub cluster does not contain the same number of examples.

- **Disadvantages**
    - The main drawback of this algorithm, like most oversampling techniques is the possibility of over-fitting the training data.

Algorithmic ensemble techniques are another type of techniques which main purpose is to improve the performance of single classifiers. The approach involves constructing several two stage classifiers from the original data and then aggregate their predictions.

3.2.3.4 Bagging

Bagging is an abbreviation of Bootstrap Aggregating. The conventional bagging algorithm involves generating 'n' different bootstrap training samples with replacement. And training the algorithm on each bootstrapped algorithm separately and then aggregating the predictions at the end [29,32,33].

Bagging is used for reducing Overfitting in order to create strong learners for generating accurate predictions. Unlike boosting, bagging allows replacement in the bootstrapped sample [29,32,33].

- **Advantages**
    - Improves stability & accuracy of machine learning algorithms.
    - Reduces variance.
    - Overcomes overfitting.
    - Improved misclassification rate of the bagged classifier.
    - In noisy data environments bagging outperforms boosting.
- **Disadvantages**
    - Bagging works only if the base classifiers are not bad to begin with. Bagging bad classifiers can further degrade performance.

3.2.3.5 Boosting

Boosting is an ensemble technique to combine weak learners to create a strong learner that can make accurate predictions. Boosting starts out with a base classifier / weak classifier that is prepared on the training data [32,33].

The base learners / Classifiers are weak learners i.e. the prediction accuracy is only slightly better than average. A classifier learning algorithm is said to be weak when small changes in data induce big changes in the classification model. In the next iteration, the new classifier focuses on or places more weight to those cases which were incorrectly classified in the last round [32,33].

Conclusion

In this chapter, we have learned about the concept of data imbalance as well as different techniques for handling this issue. No technique is flawless. All techniques have their own advantages and disadvantages. Finding out the existence of data imbalance is the initial task. The imbalance ratio is one of the measuring tools which is easy to understand and calculate. Moreover, the necessity of handling the data imbalance issue has shown with proper example. Overall, a reader should have a basic concept of what data imbalance issue is, the necessity and the technique of handling after the completion of this chapter.

Chapter 4
Feature Extraction

Introduction

Feature extraction is one of the most important phase for building a classifier. It is a process of converting inputs into features that represents the input. It's very useful for handling the string data. In this chapter, we will focus on only the string data as the protein sequences are represented as strings. As most of the classifier take numeric as input, to fed string data into classifier, we need to represent the input strings into mathematics that must represent the input data but in a different form of numeric. There are different way of doing this transformation. This chapter will cover several feature extraction techniques in the later sections.

4.1 Concept of feature extraction

Feature extraction is the process of transforming the input data into a set of features which can very well represent the input data. Features are distinctive properties of input patterns that help in differentiating between the categories of input patterns. There is no exact number of features that is necessary to successfully run the classification. Using the information of feature extraction, the classifier or the predictor provides a discriminant function which is used for separating between the classes. The quality of the classifier depends on the discriminant function. If the discriminant function can separate the positive and negative classes well, it can be considered as a good discriminant function. Most importantly, finding out the good discriminant function mostly depends on the feature extraction technique.

Let's consider a feature vector, V

$$V=\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ X_n \end{bmatrix}$$

Vector V contains n features, from X_1 to X_n. The cardinality of the vector V is called the dimentionality of the feature vector. The cardinality is the number of total elements in the vector, in our case it is n. If n is equals to 10, the vector is said to be 10 dimensional. Human can perceive 3 dimensional at most. Dimensions above three can only be represented in mathematics. To perceive higher dimensional data, sophisticated dimentionality reduction tools can be used.

4.2 Feature extraction techniques

Prediction of post-translational modification is a binary class problem. For example, development of a predictor that predicts whether a protein sequence is glycated or not. The classifier can only handle real-valued vector but not a sequence of characters. In bioinformatics, it is a crucial problem to represent the sequence of characters into a real-valued vector. Several feature extraction techniques has been developed. Some of them are shown below.

4.2.1 Amino acid composition (AAC)

Amino acid composition is one of the simplest technique among them. AAC consists of 20 component, each reflecting the fraction of one of the 20 amino acids within a protein [14,35,36]. The fraction is calculated as follows,

$$f(r) = \frac{N_r}{N}, \qquad r= 1,2,\ldots\ldots,20 \tag{4.1}$$

where, N is the length of the protein sequence and N_r is the occurrence frequency of the amino acid type r.

The calculation procedure of the AAC is simple and the extraction of AAC features is computationally much more tractable than other feature extraction approaches. However, it loses the sequence order information. Therefore, to improve the prediction performance, we have to incorporate it via other protein features.

- **Limitation:** It calculates the fractions considering the complete sequence where the sequence information is ignored. AAC does not preserve any information regarding the sequence.

4.2.2 Dipeptide Composition

To overcome the limitation of amino acid composition to some extent, Dipeptide Composition (DC) represents the fraction of every two consecutive amino acid residues of a protein [14,37,38]. So if 20 amino acids are considered, the feature vector will be 20×20 = 400 dimentional. If we consider the missing amino acids or the amino acids required for ensuring the uniformity of length as a dummy residue "X", then total number of amino acids will be 21. So the dimentionality of feature vector will be 21×21 = 441. The dipeptide composition descriptor is described as follows,

$$f(r,s) = \frac{N_{rs}}{N-1}, \quad r,s= 1,2,\ldots..,21 \tag{4.2}$$

Where N_{rs} is the co-occurrence frequency of the dipeptide denoted by amino acid type r and type s.

Dipeptide composition attempts to extract information about amino acid composition along the local order of amino acids. An advantage of DC over amino acid composition is that it uses some sequence-order information.

- **Limitation:** It considers only two local sequence-order information.

4.2.2 Tripeptide composition

Tripeptide Composition represents the fraction of every three consecutive amino acid residues of a protein, tried to overcome the limitation of dipeptide composition [39]. So if 20 amino acids are considered, the feature vector will be $20\times 20 \times 20$ = 8000 dimensional. If we consider the missing amino acids or the amino acids required for ensuring the uniformity of length as a dummy residue "X", then total number of amino acids will be 21. So the dimentionality of feature vector will be $21\times 21 \times 21$ = 9261. The dipeptide composition descriptor is described as follows,

$$f(r,s,t) = \frac{N_{rst}}{N-2},\ r,s,t= 1,2,\ldots..,21 \quad (4.3)$$

Where N_{rst} is the co-occurrence frequency of the tripeptide denoted by amino acid type r s and t.

Tripeptide composition attempts to extract information more than dipeptide composition preserving sequence order information of length 3. While preserving, it end up with a higher dimensional feature vectors.

- **Limitation:** Higher dimensional feature vector slows the learning rate compared to dipeptide and amino acid composition making the tripeptide computationally less tractable.

4.2.3 Sequence-order-coupling number

Sequence order coupling number is a feature extraction technique based on the quasi-sequence-order descriptors are proposed by Chou (2000) [40]. They are derived from the distance matrix between the 20 amino acids. Lag is a term defined as the distance between to amino acid residue. Maxlag is the maximum lag and the length of the protein must be not less than maxlag.

The d-th rank sequence-order-coupling number is defined as:

$$T_d = \sum_{i=1}^{N-d} (d_{i,i+d})^2 \qquad d = 1,2,\ldots\ldots, \text{maxlag} \quad (4.4)$$

Where $d_{i,i+d}$ is the distance between the two amino acids at position i and i + d. The sequence order coupling number has a parameter nlag which refers to the maxlag in the equation which can be tuned for optimal performance. The distance is the predefined chemical distance taken from the Scheneider-Wrede physicochemical distance matrix (Schneider and Wrede, 1994) and Grantham (1974) chemical matrix. These matrix are constructed from the chemical property of each amino acids. If we use both of the matrices, the total dimension of the feature vector will be nlag*2.

- **Limitation:** As it requires chemical distance, there is no chemical distance of dummy residues. So Sequence-order-coupling number does not work for the sequence with dummy residues.

4.2.4 Pseudo-amino acid composition (PseAAC)

This group of descriptors are proposed by Chou [38,41,42]. PseAAC descriptors are also named as the type 1 pseudo-amino acid composition. Let $H_1^0(i), H_2^0(i), M^0(i)$ (i=1,2,3,…,20) be the original hydrophobicity values, the original hydrophilicity values and the original side chain masses of the 20 natural amino acids, respectively. They are converted to following qualities by a standard conversion.

$$H_1(i) = \frac{H_1^0(i) - \frac{1}{20}\sum_{i=1}^{20}(i)}{\sqrt{\frac{\sum_{i=1}^{20}[[H_1^0(i) - \frac{1}{20}\sum_{i=1}^{20} H_1^0(i)]^2}{20}}} \quad (4.5)$$

$H_2^0(i)$ and $M^0(i)$ are normalized as $H_2(i)$, M (i) in the same way.

In the figure 3.1, panel (a) reflects the correlation mode between all the most contiguous residues called the first-tier., panel (b) that between all the second-most contiguous residues, called the second-tier and panel (c) that between all the third-most contiguous residues, called the third-tier. Then, a correlation function can be defines as,

$$\Theta(R_i, R_j) = \frac{1}{3}\left\{[H_1(R_i) - H_1(R_j)]^2 + [H_2(R_i) - H_2(R_j)]^2 + [M(R_i) - M(R_j)]^2\right\} \quad (4.6)$$

This correlation function is actually an average value for the three amino acid properties: hydrophobicity value, hydrophilicity value and side chain mass. Therefore, we can extend this definition of correlation functions for one amino acid property or for a set of n amino acid properties.

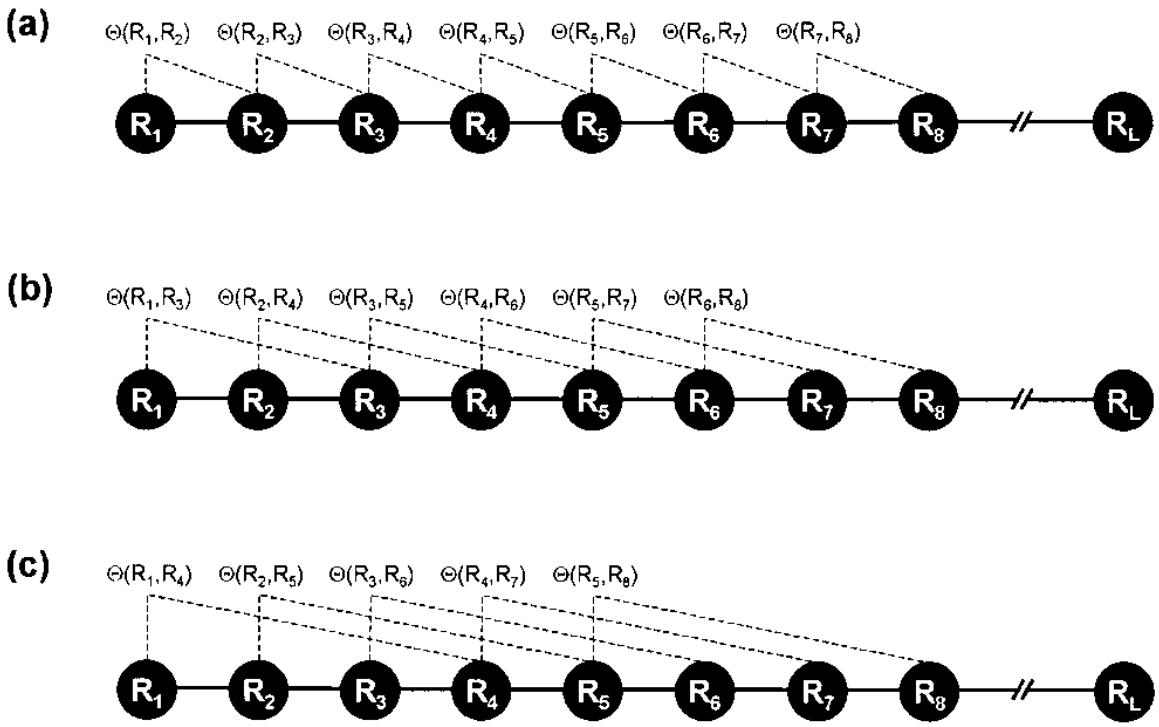


Figure 4.1 A schematic drawing to show (a) the first-tier, (b) the second-tier, and (3) the third-tier sequence order correlation mode along a protein sequence [43]

In the figure 3.1, panel (a) reflects the correlation mode between all the most contiguous residues called the first-tier., panel (b) that between all the second-most contiguous residues, called the second-tier and panel (c) that between all the third-most contiguous residues, called the third-tier.

Then, a correlation function can be defines as,

$$\Theta(R_i, R_j) = \frac{1}{3}\left\{[H_1(R_i) - H_1(R_j)]^2 + [H_2(R_i) - H_2(R_j)]^2 + [M(R_i) - M(R_j)]^2\right\} \quad (4.7)$$

This correlation function is actually an average value for the three amino acid properties: hydrophobicity value, hydrophilicity value and side chain mass. Therefore, we can extend this definition of correlation functions for one amino acid property or for a set of n amino acid properties.

For one amino acid property, the correlation can be defined as:

$$\Theta(R_i, R_j) = [H_1(R_i) - H_1(R_j)]^2 \quad (4.8)$$

Where, $H_1(R_i)$ is the amino acid property of amino acid R_i after standardization.

For a set of n amino acid properties, it can be defined as:

$$\Theta(R_i, R_j) = \frac{1}{n} \sum_{k=1}^{n} [H_k(R_i) - H_k(R_j)]^2 \quad (4.9)$$

here $H_k(R_i)$ is the k-th property in the amino acid property set for amino acid R_i.

A set of descriptors named sequence order-correlated factors are defined as:

$$\begin{aligned} \Theta_1 &= \frac{1}{N-1} \sum_{i=1}^{N-1} \Theta(R_i, R_{i+1}) \\ \Theta_2 &= \frac{1}{N-2} \sum_{i=1}^{N-2} \Theta(R_i, R_{i+2}) \\ \Theta_3 &= \frac{1}{N-3} \sum_{i=1}^{N-3} \Theta(R_i, R_{i+3}) \\ &\dots \\ \Theta_\lambda &= \frac{1}{N-\lambda} \sum_{i=1}^{N-\lambda} \Theta(R_i, R_{i+\lambda}) \end{aligned} \quad (4.10)$$

Lambda λ is parameter to be specified which must be less than the Length of protein sequence, L. Let f_i be the normalized occurrence frequency of the 20 amino acids in the protein sequence, a set of 20+ λ descriptors called the pseudo-amino acid composition for a protein sequence can be defines as,

$$X_c = \frac{f_c}{\sum_{r=1}^{20} f_r + w \sum_{j=1}^{\lambda} \theta_j} \quad \text{for } (1 \leq c \leq 20)$$

$$X_c = \frac{w\theta_{c-20}}{\sum_{r=1}^{20} f_r + w \sum_{j=1}^{\lambda} \theta_j} \quad \text{for } (21 \leq c \leq 20+\lambda) \quad (4.11)$$

where w is the weighting factor for the sequence-order effect as suggested by Kuo-Chen Chou. The dimensional of the feature vector is equal to the $20+\lambda$ where $\lambda < L$.

4.2.5 Peptide sequence evolution based feature extraction

Protein sequences go through several random evolutions. This evolution includes single or multiple alternations, insertions or deletions of amino acids. In results, these create completely new sequences, though these came from the same origin [44,45]. To find these similarity among the protein/peptide sequences, a sophisticated tool named BLAST (Basic Local Alignment Search Tool) which finds the resemblance of a query sequence with the database of sequences.

PSI-BLAST [46] (Position-Specific Iterative Basic Local Alignment Search Tool) provides a PSSM [46] (Position-Specific Scoring Matrix) for each query sequence. PSI-BLAST is tool, designed by NCBI (National Center of Bioinformatics Information), takes two parameters. One is number of iteration, because PSI-BLAST [46] is a iterative proves, and another is the e-value threshold. At first iteration it takes peptide sequence which is applied on the database and calculate e value for each peptide or protein sequence from the database. The e-value less than threshold is considered homogeneous, came from the same origin, and these homogeneous peptide/protein sequence is used to generate PSSM matrix which is used form the next iterations instead of query sequence. Then the last generated PSSM [20] matrices of all query protein sequences are further processed to represent as feature vector which will eventually be fed into classifier.

The typical sequential evolutionary feature extraction technique uses complete database of protein sequences. In contrast, we have created a custom database from the peptide sequences of our dataset which gives an acceptable performance as well as eliminates the necessity of the premade protein database. The PSI-BLAST [46] is run against the custom made database where each peptide sequence is treated individually as a query sequence which provides PSSM [46] like matrix as an output. Further processing is done to extract the features from these matrices. This method is named as peptide sequence evolution based

feature because it takes only peptide sequences to make custom database instead of considering the whole protein database. Let's see the graphical differences between the typical sequential evolution method and the peptide sequence evolution based feature extraction method.

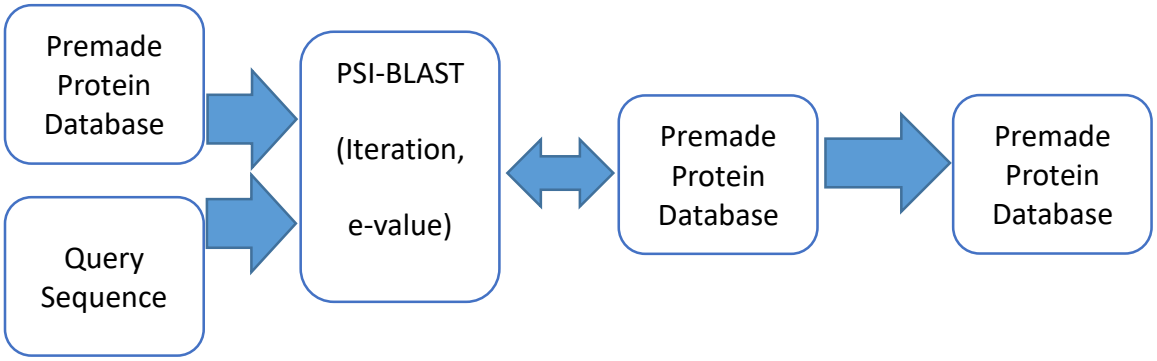


Figure 4.2 Typical sequential evolutionary feature extraction technique

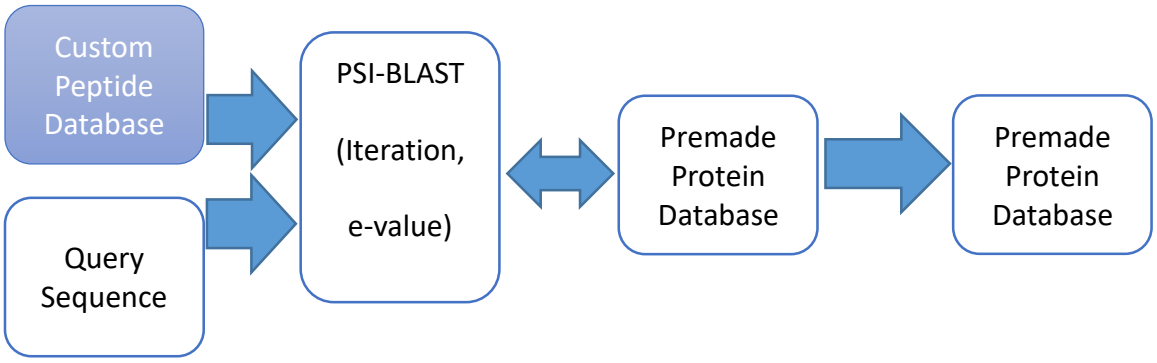


Figure 4.3 Proposed peptide sequence evolution based features

The following technique is used to generate feature vector from the dataset.

a) A query peptide sequence is denoted by P and can be represented as

$$P= R_1R_2R_3R_4R_5 \dots \dots \dots R_L$$

From the study of Schaffer et al. [46] we know that, the information of sequential study evaluation of P, can be represented by a 20*L matrix as following equation.

$$\begin{bmatrix} \dot{E}_{1\to1} & \dot{E}_{2\to1} & \dots\dots & \dot{E}_{L\to1} \\ \dot{E}_{1\to2} & \dot{E}_{2\to2} & \dots\dots & \dot{E}_{L\to2} \\ \vdots & \vdots & & \vdots \\ \dot{E}_{1\to20} & \dot{E}_{2\to20} & \dots\dots & \dot{E}_{L\to20} \end{bmatrix} \quad (4.12)$$

Where, 20 refers to the 20 different amino acids ordered by alphabetically, L refers to the length of P, $\dot{E}_{i\to j}$ denotes the propensity of the i-th amino acid residue being mutated to the j-th amino acid at the time of evaluation process. The custom database has been created by all the peptide sequence in the benchmark dataset [15] and each sequence is individually treated as a query sequence against the custom database. The search method used two iterations and 0.001 as the E-value cutoff.

b) The new matrix can be derived from the matrix in equation (5) as follows,

$$\begin{bmatrix} E_{1\to1} & E_{2\to1} & \dots\dots & E_{L\to1} \\ E_{1\to2} & E_{2\to2} & \dots\dots & E_{L\to2} \\ \vdots & \vdots & & \vdots \\ E_{1\to20} & E_{2\to20} & \dots\dots & E_{L\to20} \end{bmatrix} \quad (4.13)$$

With

$$E_{i\to j} = \frac{\dot{E}_{i\to j} - \bar{\dot{E}}_j}{SD(\bar{\dot{E}}_j)} \quad i= 1,2,\dots..,L \; ; j= 1,2,\dots, 20 \quad (4.14)$$

Where

$$\bar{\dot{E}}_j = \frac{1}{L}\sum_{i=1}^{L} \dot{E}_{i\to j} \qquad j= 1,2,\dots..,20 \quad (4.15)$$

Here $\bar{\dot{E}}_j$ refers to the mean of $\dot{E}_{i\to j}$ for i= 1,2,…20 and the standard deviation is denoted and defined by the following equation,

$$SD(\bar{\dot{E}}_j) = \sqrt{\sum_{i=1}^{L} \left[\dot{E}_{i\to j} - \bar{\dot{E}}_j\right]^2 / L} \quad (4.16)$$

c) The new matrix MM^T is measured by multiplying M with the transpose of M that becomes (20×20) matrix of 400 elements. Moreover the resultant matrix is a symmetric matrix contains 210 unique information, 20 comes from the diagonal and 190 = (400-20)/2 elements from lower or upper triangle matrix. In our study, the lower triangular matrix with the diagonal has been considered, as follows,

$$\begin{bmatrix} (1) & & & \\ (2) & (3) & & \\ (4) & (5) & (6) & \\ : & : & : & \\ (191) & (192) & (193) \ldots . & (210) \end{bmatrix} \tag{4.17}$$

The matrix of equation converted into vector representation of 210 elements as show below,

$$P_{evo} = [\theta_1^E \theta_2^E \ldots \theta_u^E \ldots \theta_{210}^E]^T \tag{4.18}$$

Conclusion

In this chapter, we have learnt about what feature extraction is, necessity of feature extraction and different feature extraction technique. There are other several techniques available but here we have discussed a small portion in brief. The other techniques are not considered as only these techniques mentioned in this chapter has been implemented. For this reason, other technique have not been discussed here.

Chapter 5

Support Vector Machine Classifier

Introduction

Classification is the process of associating each data objects into classes. There can be two or more classes. Two class problems are called binary classification problems. More than two class problems are called multiclass classification problems. The output of a classification is a discriminant function which separates each classes. The post-translational modification problem is the binary classification problem as there is only two classes. So this chapter will cover only the methods of binary class problems. More specifically this chapter will cover detailed view of support vector machine classifier because it is implemented in the chapter of result and discussions.

5.1 Support vector machine (SVM)

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

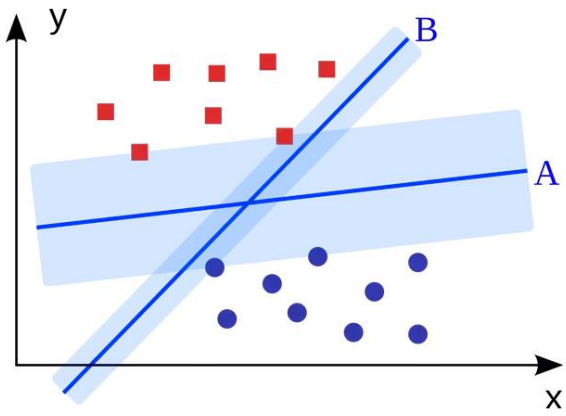


Figure 5.1 Two valid classifying line with different margin width [47]

The figure 5.1 contains two discriminating line A and B. Both of them are separating two classes without any error. So now the question is, which line is the best classifier. To find

the answer of the classifier we need to find a measuring tool for calculating the goodness. First proposal was made to consider the line with fat or wider margin is regarded to be good compared to the line with narrow margin. The logic behind that, the wider margin is less susceptible to the change of test dataset compared to the narrower margin. So for classification, the first constrain is to maximize the width of the margin. The following section contains the mathematics of support vector machine with the constraint to maximize the margin.

5.2 Mathematics of SVM

Support Vector Machine (SVM) is a binary classifier that gives the optimal hyper plane for separating the classes by the margin with maximum width. So the constraint problem becomes as follows [48,49],

$$\text{Minimize}_{\ w,b}\ \frac{1}{2}\|W\|^2 \tag{5.1}$$

$$\text{Subject to}\ \ y_i(w^T x_i + b) \geq 1, \quad i= 1,2,3,\ldots..,n$$

A penalty term is added to allows errors for finding the wider margin.

$$\text{Minimize}_{\ w,b}\ \frac{1}{2}\|W\|^2 + C\sum_{i=1}^{n}\xi_i \tag{5.2}$$

$$\text{Subject to} \qquad y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i= 1,2,3,\ldots..,n$$

$$\xi_i \geq 0, \quad i= 1,2,3,\ldots..,n$$

From the Lagrange multipliers, the dual formulation is obtained and represented with respect to $\alpha_i$variable. [47,48].

$$\text{Maximize}_{\ \alpha}\ \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\ \sum_{j=1}^{n}\alpha_i\,\alpha_j y_i y_j x_i^T x_j \tag{5.3}$$

$$\text{Subject to}\ \sum_{i=1}^{n} y_i\alpha_i = 0\ ,\ \ 0 \leq\ \alpha_i \leq C$$

$$\text{For } i = 1,2,3,\ldots..,n$$

The final form of linear discriminant function as follows,

$$F(x) = \sum_{i}^{n} y_i\alpha_i x_i^T x\ + b \tag{5.4}$$

To make the linear function as nonlinear, we need to use a nonlinear function Ø: X→ F, which maps from input space X to feature space F. Now the form of the optimization function becomes as follows using the kernel function [48-50],

$$\text{Maximize}_{\ \alpha}\ \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\ \sum_{j=1}^{n}\alpha_i\,\alpha_j y_i k(x_i, x_j) \qquad (5.5)$$

$$\text{Subject to } \sum_{i=1}^{n} y_i\alpha_i = 0\ ,\ \ 0 \le\ \alpha_i \le C$$

$$\text{For } i = 1,2,3,\ldots..,n$$

In term of kernel function, the discriminant function becomes,

$$F(x) = \sum_{i}^{n} y_i\alpha_i k(x_i, x_j) + b \qquad (5.6)$$

5.3 Regularization term (C term)

We already have noticed the C parameter of the SVM in the earlier section of mathematics of SVM. This parameter is called regularization term or the error term which tradeoffs between misclassification and overfitting. If we want zero tolerance about misclassification, the overfitting problem can occur. The relationship between misclassification and overfitting is inversely proportional to each other. The following figures shows this tradeoff.

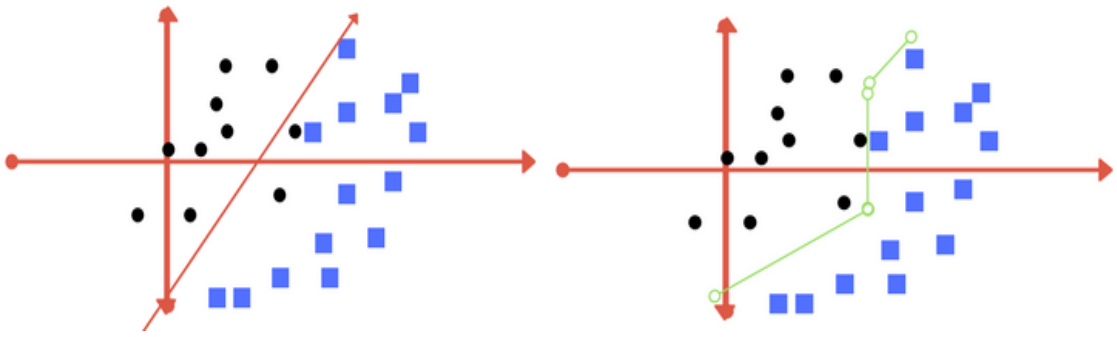


Figure 5.2 Tradeoff between misclassification and overfitting [47]

The figure 5.2 contains two images. Left image has a separating line that is simpler than the right image. Left image allows some error to occur but the right image has zero tolerance regarding errors. The right image faces overfitting problem but the left image does not faces overfitting problem. Left image is just fit. The overfitting problem is the

problem when the predictor or classifier performs well in training set but the performance for the test set is poor. Though right image does not allow any error to occur in the training set, but misclassifies for the test set as it is more susceptible to the change in the dataset compared to the left image.

- Very large value of C causes overfit, works well for the training set but not for the test set.
- Very small value of C causes underfit, does not work well for both training set and test set
- Balanced C results in just fit, works well for both training and test set.

For finding the best value of C, the cross-validation method is used. This method will be discuss in the future articles.

5.4 Kernel function

SVM algorithms use a set of mathematical functions that are defined as the kernel. The function of kernel is to take data as input and transform it into the required form. Different SVM algorithms use different types of kernel functions. These functions can be different types. Kernel function can be several types, such as linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.

5.4.1 Linear kernel

Linear kernel is the dot product which works well for the linearly separable data [49]. For example, text processing because text is often linearly separable. The most important factor for linear kernel is that it is faster than any other kernel. Another thing is that, it takes less parameter compared to other kernel functions. Major limitation is that it works only for linearly separable data. The following picture is a visualization of text processing where both RBF and linear has been used. Both kernel worked very well for linearly separable data but for the reason of linear kernel being fast, it’s wiser to use linear kernel rather than RBF kernel for the text processing.

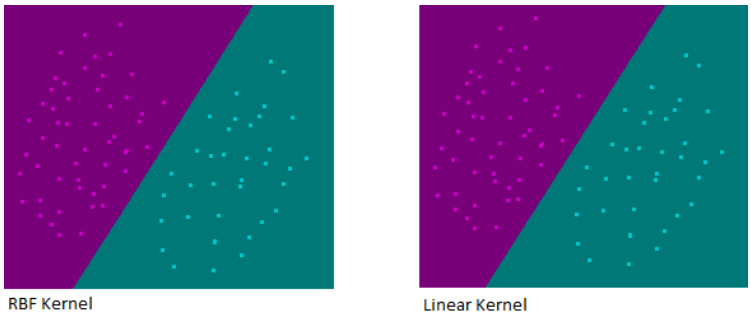


Figure 5.3 RBF kernel vs Liner kernel for text processing [51]

5.4.2 Polynomial kernel

Polynomial kernel is the nonlinear kernel which also works for linearly non-separable data. The kernel takes d as a parameter where d means the degree of the polynomial function. Very large degree tends to overfit the model. Here tuning d degree is an additional parameter which ads versatility of polynomial kernel while designing a predictor. The mathematics of polynomial function follows [49,51],

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^d \tag{5.7}$$

Where d is the degree of the polynomial. Polynomial functions are popular in image processing and natural language processing. The degree-2 is commonly used as higher degree tends to overfit the model.

5.4.3 Radial basis function kernel (RBF kernel)

Radial basis function kernel is the most popular kernel used for SVM. The popularity has reasons for having a tuning parameter sigma.

The radial basis function (RBF) is defined as [49,51],

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i, x_j\|^2}{2\sigma^2}\right) \tag{5.8}$$

Here, σ is the width for the kernel function.

$$k(x_i, x_j) = \exp\left(-\gamma \|x_i, x_j\|^2\right) \quad (5.9)$$

For $\gamma > 0$

Where $\gamma = ½\sigma^2$

The γ, gamma parameter can be tuned for radial basis function which is inversely proportional to the variance, σ^2. Higher gamma means the nearby points has influence on the separating plane. Lower gamma means the far away points also have the influence on the separating plane.

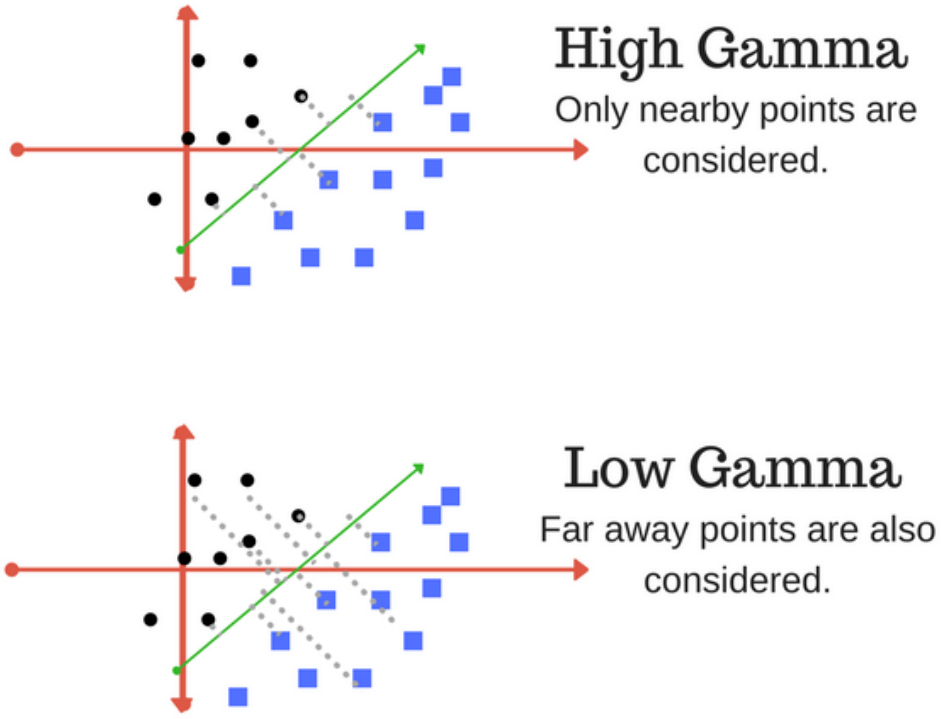


Figure 5.4 Effects of value of gamma in the SVM classification [47]

5.5 Model selection

For getting the best parameter value for classifier or the predictor, three most used methods are, independent dataset test, k-fold cross-validation (subsampling) and jackknife test Jackknife test estimates with (n-1) sampling leaving one sample for testing. This estimates gives same result every time it runs for a specific dataset but the limitation is, it runs 'n' times for which the computational time is very high for large dataset. In contrast, k-fold

cross-validation takes very low computational time for large dataset compared to jackknife testing. Datasets can be three types,

- **Test set**

  The model is initially fit on a training dataset that is a set of examples used to fit the parameters of the model. The model is trained on the training dataset using a supervised learning method.

- **Validation set**

  The fitted model is used to predict the responses for the observations in a second dataset called the validation dataset.

- **Test set**

  The test dataset is a dataset used to provide an unbiased evaluation of a final model fit on the training dataset. When the data in the test dataset has never been used in training (for example in cross-validation), the test dataset is also called a holdout dataset.

5.5.1 Independent dataset test

In the independent dataset test the model is fit with training set and validation set and measure the performance with validation set [28,52,53]. The complete dataset is divided into three parts. One part is kept aside which does not perform any action for building the model or classifier. That part is called the independent test set. The model is fit with two datasets. One is called training dataset which is used for training and another is called the validation set which is used to find out the best parameters of the model. After getting the parameters of newly built model, the model's performance is measures using the independent dataset.

- **Advantage**
  - Performance measure does not exhibit any bias result because the test is separated from the training and validation set.

5.5.2 Jackknife test

Jackknifing, which is similar to bootstrapping, is used in statistical inference to estimate the bias and standard error in a statistic, when a random sample of observations is used to calculate it. The basic idea behind the jackknife estimator lies in systematically recomputing the statistic estimate leaving out one observation at a time from the sample set. From this new set of "observations" for the statistic an estimate for the bias can be calculated, as well as an estimate for the variance of the statistic.

If the number of observation is n, it considers (n-1) observations as test set while leaving one as validation set. This process takes total of n times while each observations is considered a validation set for each time [28,52,53].

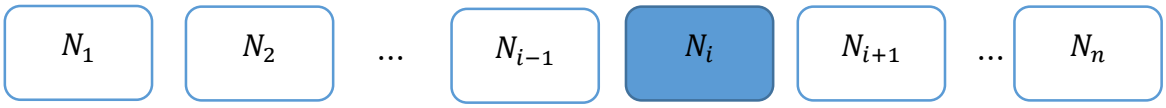


Figure 5.5 Single Iteration of Jackknifing when N_i is considered as validation data

In the figure 5.5, the solid blue box is considered as the validating observation and the rest is considered to be training set for this iteration. There will be n iterations for the n number of observations.

- **Advantages**
  - Simple technique.
  - Works well for small datasets.
- **Limitation**
  - If the number of observations increases, it becomes computationally intractable.

5.5.2 K-fold cross validation

In k-fold cross-validation, the original sample is randomly partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k − 1 subsamples are used as training data. The cross-validation process is then repeated k times, with each of the k subsamples used exactly once as the validation data [28,52,53]. The k results can then be averaged to produce a

single estimation. The advantage of this method over repeated random sub-sampling (see below) is that all observations are used for both training and validation, and each observation is used for validation exactly once. 10-fold cross-validation is commonly used, but in general k remains an unfixed parameter.

For example, setting k = 10 results in 10-fold cross-validation. In 10-fold cross-validation, 9 folds are treated as training dataset and only one fold is treated as validation set.

F1 F2 F3 F4 F5 F6 F7 F8 F9 | F10

Figure 5.6 Single iteration of 10-fold cross validation

In the figure 5.6, each fold is denoted by Fi, for $1 \leq i \leq 10$, The folding is an iteration process of 10 iteration. On each iteration one fold is considered as validation set and another folds are considered as training sets. In the figure, F10, denoted by green color, is the validation set and from F1 to F9, denoted by blue color, is the training sets.

When k = n (the number of observations), the k-fold cross-validation is exactly the leave-one-out cross-validation.

- **Advantage**
  - Works well for large datasets.
- **Limitations**
  - K term needs to be specified depending on the datasets.
  - Folding of datasets need close observation for multiclass problem.

Conclusion

In this chapter, we have learnt about what support vector machine is and the mathematical derivation of the SVM. We also learnt about the parameters taken by the support vector machine and the way of getting the best possible parameters for the proposed model. There are other techniques beside support vector machine for classification. For example, random forest and naïve bayes. But we limited the chapter within the support vector machine because we have implemented the support vector machine only in our proposed method.

Chapter 6
Measuring Matrices

Introduction

It is very important to measure the performance of the statistical classifier for both understanding how well it performs as well as to compare with the existing model. Confusion matrices are the measurement tools for measuring the performance of a predictor or classifier. Sometimes the confusion matrix are also called the error matrix which is a specific table layout that allows visualization of the performance of a supervised algorithm. For unsupervised algorithms, the performance measuring tool is called the matching matrix. As we are discussing about the supervised methods, this chapter will cover the confusion matrices only. In this chapter, we will learn several measuring matrices and their usage and limitations.

6.1 Table of confusion

In predictive analytics, a table of confusion (sometimes also called a confusion matrix), is a table with two rows and two columns that reports the number of false positives, false negatives, true positives, and true negatives. This allows more detailed analysis than mere proportion of correct classifications (accuracy). Accuracy is not a reliable metric for the real performance of a classifier, because it will yield misleading results if the data set is unbalanced.

	Actual class	
Predicted class	True positives	False positives
	False negatives	True negatives

Figure 6.1 Table of confusion

- A true positive is an outcome where the model correctly predicts the positive class.

- A true negative is an outcome where the model correctly predicts the negative class.
- A false positive is an outcome where the model incorrectly predicts the positive class.
- A false negative is an outcome where the model incorrectly predicts the negative class.

True Positive (TP): Actual: Yes. Predicted: Yes.	False Positive (FP): Actual: No. Predicted: Yes.
False Negative (FN): Actual: Yes. Predicted: No.	True Negative (TN): Actual: No. Predicted: No.

Figure 6.2 2×2 confusion matrix with conditions

6.2 Different measuring matrices

This section contains several techniques for measuring the performance.

6.2.1 Accuracy

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition [54-57].

$$\text{Accuracy} = \frac{\text{No of correct predictions}}{\text{Total number of predictions}}$$

For binary classification,

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{6.1}$$

- **Limitation**
  - Accuracy is not a good measure working with a class-imbalanced data set.

6.2.2 Precision

Precision shows the proportion of positive identifications was actually correct [54-57].. Precision is also called by positive predictive value (PPV). For calculating the precision, the following formula has been used.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{6.2}$$

6.2.3 Recall / Sensitivity

Recall/ Sensitivity shows the proportion of actual positives was identified correctly [54-57]. Recall/ Sensitivity is also called by true positive rate (TPR). Sensitivity quantifies the avoiding of false negatives. For calculating the recall, the following formula has been used.

$$\text{Recall / Sensitivity} = \frac{TP}{TP+FN} \tag{6.3}$$

6.2.4 Specificity

Specificity measures the proportion of actual negatives that are correctly identified [54-57]. Specificity sometimes is called by true negative rate (TNR). Specificity quantifies the avoiding of false positives. For calculating the specificity, the following formula has been used.

$$\text{Specificity} = \frac{TN}{TN+FP} \tag{6.4}$$

6.2.4 F-score

In statistical analysis of binary classification, the F1 score (also F-score or F-measure) is a measure of a test's accuracy [54-57]. It considers both the precision p and the recall r of the test to compute the score. The traditional F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall. The formula of calculating F-score is shown below,

$$\text{F-score} = \frac{2}{\frac{1}{\text{Recall}}+\frac{1}{\text{Precision}}} \tag{6.5}$$

$$= \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

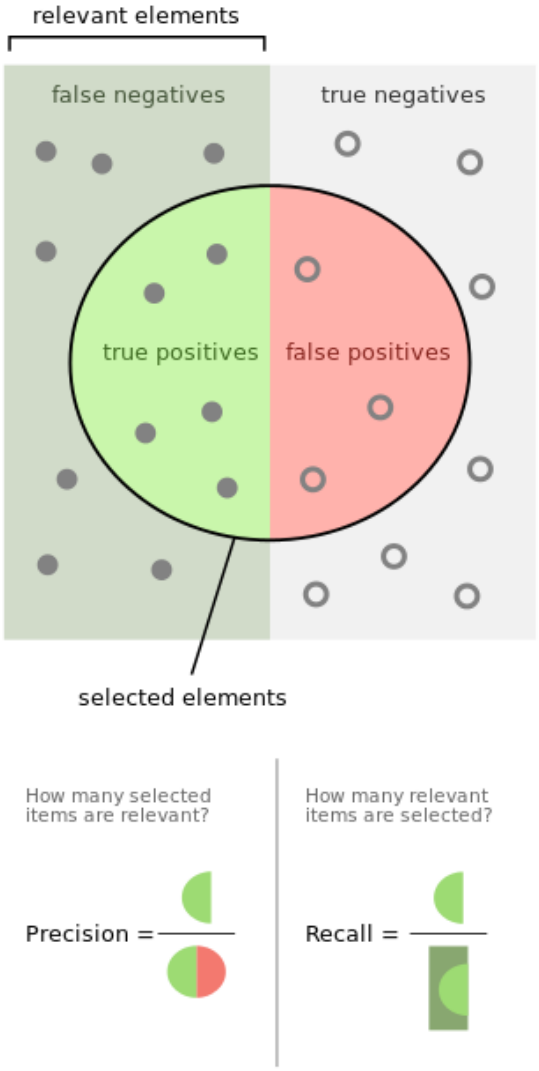


Figure 6.3 Graphical representation of Precision and Recall [58]

6.2.5 Matthews correlation coefficient

The Matthews correlation coefficient is used in machine learning as a measure of the quality of binary (two-class) classifications, introduced by biochemist Brian W. Matthews in 1975. It 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 [54-57].

$$\mathrm{MCC} = \frac{\mathrm{TP}\times\mathrm{TN}-\mathrm{FP}\times\mathrm{FN}}{\sqrt{(\mathrm{TP}\times\mathrm{FP})(\mathrm{TP}\times\mathrm{FN})(\mathrm{TN}\times\mathrm{FP})(\mathrm{TN}\times\mathrm{FN})}} \tag{6.6}$$

6.2.6 ROC curve

A receiver operating characteristic curve (ROC curve) is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, recall or probability of detection in machine learning. The false-positive rate is also known as the fall-out or probability of false alarm and can be calculated as (specificity) [54-57]..

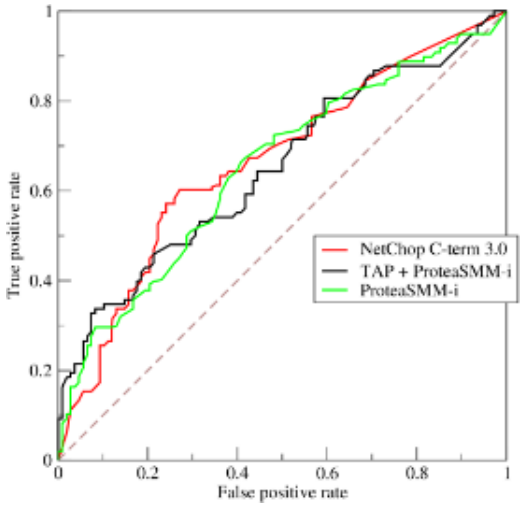


Figure 6.4 ROC curve of three predictors of peptide cleaving in the proteasome [59]

Another interpretation called AUC, area under curve, is used for comparing the models. As in the figure 6.4, we cannot decide which method is performing well for the overlapping of three curves. So the idea is to take the total area under each curve by integration. The method with largest area is called as the best performing method among the rest. This is a measurement of overall performance. The AUC does not ensure that the curve with maximum area will always perform best in each case.

Conclusion

In this chapter, we have discussed about the how the performance of a model can be calculated. This gave us an idea of comparing different models. We also discussed about some of the measuring matrices briefly. Each measuring matrices represents different characteristics of the model. Some measuring matrices are susceptible to data imbalance issue, for example accuracy. No measuring matrices carries complete picture of performance. So a set of measuring matrices are used for getting an overall picture of the performance. Moreover, this chapter gave us an idea of how to compare the newly developed tools against the existing tools.

Chapter 7

Results and performance analysis

Introduction

The previous chapters contain all the necessary informations for the implementations. From the idea of previous chapter, a proposed method is implemented. We will find out how our proposed methods performed against the existing tools. First step is to select the database. Then the preprocessing is done. After that, an efficient feature extraction technique has been chosen or implemented. At last the predictor is built by feeding the extras acted feature into the classifier and tuning the parameters. In this chapter, we will go through all the steps sequentially for getting the idea about how the new model is designed and the performance of the newly built model compared to the previously developed models.

7.1 Dataset selection

PLMD (Protein Lysine Modifications Database) is an online data resource specifically designed for protein lysine modifications (PLMs). The PLMD data resource consists all the modifications that occurs for lysine residue of protein. It contains the proteins of human as well as other proteins.

In the dataset selection we will include, two datasets. First one is hydroxylation which is taken from PLMD and another one of glycation which is taken from the base papers. The hydroxylation dataset is taken for the comparative performance of different feature extraction techniques and the glycation dataset is taken for building the actual predictor. The glycation dataset is taken from the base paper for comparing the results with the previously developed tools.

Sources:

- For Hydroxylation: http://plmd.biocuckoo.org/download.php
- For Glycation: http://123.206.31.171/BPB_GlySite/data.html

Table 7.1 Hydroxylation dataset sample

PLMD ID	Uniprot Accession	Position	Type	Sequence	Species	PMIDs
PLMD-694	O18495	30	Hydroxylation	MQMKATILILVALFMIQ….	Styela clava	10978343
PLMD-2988	P02452	265	Hydroxylation	MFSFVDLRLLLLLAATA……	Homo sapiens	4319110
PLMD-2989	P02453	264	Hydroxylation	MFSFVDLRLLLLLAATA….	Bos taurus	1164916
PLMD-2991	P02457	254	Hydroxylation	MFSFVDSRLLLLIAATV….	Gallus gallus	1165248
PLMD-3041	P02745	100	Hydroxylation	MEGPRGWLVLCVLAISLA…	Homo sapiens	486087
PLMD-3042	P02746	110	Hydroxylation	MMMKIPWGSIPVLMLL….	Homo sapiens	486087

The dataset contains seven columns,

1. **PLMD-ID:** The unique id of protein sequence in the Protein Lysine Modification Dataset (PLMD).
2. **Uniprot Accession:** The universal protein knowledgebase for accessing each proteins.
3. **Position:** The position of the responsible residue where the modification has occurred.
4. **Type:** Type denotes the specific post-translational modification.
5. **Species:** Specifies the species on which the modification has occurred.
6. **PMIDs:** Protein modification identifier.

The specification of the dataset of hydroxylation is given below.

- The number of proteins: 32.
- Positives sites: 135
- Negative sites: 906
- Total: 1041

- Imbalance ratio: $\frac{906}{135} = 6.7$

The hydroxylation dataset needs to be preprocessed before going to preprocessing. Here preprocessing means the process of extracting of peptide sequence from protein sequence discussed at 3.2 article. Unlikely the glycation dataset is already processed. The sample of dataset of glycation is split into two parts. One contains the positive sites and the other contains the negative sites.

Table 7.2 Sample of positive sites for glycation dataset

ID	Site	Sequence
P00325	11	AGKVIKCKAAVLWEV
P00325	40	KAYEVRIKMVAVGIC
P00435	116	PNFMLFEKCEVNGEK
P00441	4	XXXXMATKAVCVLKG
P00441	10	TKAVCVLKGDGPVQG
P00441	31	KESNGPVKVWGSIKG
P00698	19	LPLAALGKVFGRCEL
P01317	53	RGFFYTPKARREVEG

Table 7.3 Sample of negative sites for glycation dataset

ID	Site	Sequence
P10591	98	KLIDVDGKPQIQVEF
P10636	557	PKSPSSAKSRLQTAP
P02647	64	FEGSALGKQLNLKLL
P16615	1003	ECVQPATKSCSFSAC
P10591	497	KGTGKSNKITITNDK
P11277	58	EREVVQKKTFTKWVN
Q03626	1204	ERPQKPTKSEGYLYT
Q03626	234	PRFGVDVKVPNAISV

Where positive sites denotes the glycated sites and negative sites denotes the non-glycated sites.

Characteristics of lysine glycation dataset,

- Responsible residue : K, Lysine
- Number of positive sites: 223
- Number of negative sites: 446

- Total sites: 696
- Imbalance ratio: 2:1
- Window size: 15

7.2 Implementation of proposed feature extraction technique

The new proposed feature extraction technique has been discussed mathematically and theoretically in the chapter 4. Now the walkthrough of a single sequence will be shown here for getting the complete idea.

Lets consider the query sequence,

AADFVESKDVCKNYA

The PSI-BLAST is applied for this sequence on the custom peptide database which gave us the following PSSM matrix of size (15×20).

Table 7.4 PSSM matrix for a specific sequence

4	-2	-2	-2	0	-1	-1	0	-2	-1	-2	-1	-1	-2	-1	1	0	-3	-2	0
4	-2	-2	-2	0	-1	-1	0	-2	-1	-2	-1	-1	-2	-1	1	0	-3	-2	0
-2	-2	-4	-4	-1	-2	-3	-4	-3	2	4	-3	2	0	-3	-3	-1	-2	-1	1
0	-3	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-3	-3	-3
0	-4	-3	-4	10	-3	-4	-3	-3	-1	-1	-3	-2	-3	-3	-1	-1	-3	-3	-1
-2	-2	-4	-4	-1	-2	-3	-4	-3	2	4	-3	2	0	-3	-3	-1	-2	-1	1
0	-3	-3	-3	-1	-2	-3	-3	-3	3	1	-3	1	-1	-3	-2	0	-3	-1	4
-1	2	0	-1	-3	1	1	-2	-1	-3	-3	5	-2	-3	-1	0	-1	-3	-2	-3
-2	-2	1	6	-4	0	2	-1	-1	-3	-4	-1	-3	-4	-2	0	-1	-5	-3	-3
-2	-2	-2	-3	-3	-2	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
-2	-3	-3	-4	-3	-4	-4	-3	-1	0	0	-3	0	7	-4	-3	-2	1	3	-1
-1	-2	-2	-2	-3	-1	-1	-2	-2	-3	-3	-1	-3	-4	8	-1	-1	-4	-3	-3
-1	0	0	2	-4	2	5	-2	0	-4	-3	1	-2	-4	-1	0	-1	-3	-2	-3
-1	-2	-2	-2	-3	-1	-1	-2	-2	-3	-3	-1	-3	-4	8	-1	-1	-4	-3	-3
0	-3	-3	-3	-1	-2	-3	-3	-3	3	1	-3	1	-1	-3	-2	0	-3	-1	4

From the PSSM matrix, a new matrix M is calculated according to the chapter 4 and the symmetric MM^T is calculated which is shown below. The results of the matrix in table 7.5 is rounded for better understanding. In actual program fractional value has been used for better accuracy.

Table 7.5 MM' matrix is calculated from PSSM matrix

14	-1	0	-1	4	1	0	6	-3	0	-3	1	-1	-4	0	9	9	-4	-4	3
-1	14	7	5	-7	11	10	-1	6	-6	-4	13	-3	-4	2	5	0	-2	-1	-6
0	7	14	12	-6	10	11	8	7	-11	-12	9	-11	-7	2	10	-3	-5	-4	-10
-1	5	12	14	-6	10	11	5	5	-8	-9	6	-8	-7	1	8	0	-7	-5	-7
4	-7	-6	-6	14	-6	-7	-2	-6	4	3	-5	2	-1	-3	0	3	-1	-3	4
1	11	10	10	-6	14	13	2	4	-8	-7	11	-6	-9	4	9	3	-7	-5	-6
0	10	11	11	-7	13	14	3	7	-9	-8	10	-7	-8	3	8	1	-6	-4	-8
6	-1	8	5	-2	2	3	14	1	-8	-9	2	-8	-5	1	9	-2	-4	-5	-6
-3	6	7	5	-6	4	7	1	14	-7	-6	5	-5	3	0	2	-7	7	9	-7
0	-6	-11	-8	4	-8	-9	-8	-7	14	12	-8	13	7	-7	-9	6	4	4	13
-3	-4	-12	-9	3	-7	-8	-9	-6	12	14	-8	13	7	-6	-11	2	5	4	10
1	13	9	6	-5	11	10	2	5	-8	-8	14	-7	-6	4	8	0	-4	-3	-8
-1	-3	-11	-8	2	-6	-7	-8	-5	13	13	-7	14	8	-8	-10	4	5	5	12
-4	-4	-7	-7	-1	-9	-8	-5	3	7	7	-6	8	14	-7	-9	-5	13	12	5
0	2	2	1	-3	4	3	1	0	-7	-6	4	-8	-7	14	3	1	-6	-5	-7
9	5	10	8	0	9	8	9	2	-9	-11	8	-10	-9	3	14	4	-7	-7	-6
9	0	-3	0	3	3	1	-2	-7	6	2	0	4	-5	1	4	14	-7	-6	8
-4	-2	-5	-7	-1	-7	-6	-4	7	4	5	-4	5	13	-6	-7	-7	14	13	2
-4	-1	-4	-5	-3	-5	-4	-5	9	4	4	-3	5	12	-5	-7	-6	13	14	3
3	-6	-10	-7	4	-6	-8	-6	-7	13	10	-8	12	5	-7	-6	8	2	3	14

As the matrix in table 7.5 contains symmetric upper triangular and lower triangular. The lower triangular matrix with diagonal (blue) is used to consider the feature vector.

So the feature vector is a (1×210) dimensional vector derived as,

[14 -1 14 0 7 14 -1 5 12 14 …… 3 14]

7.3 Implementation specifications and results

The hydroxylation dataset is used for both preprocessing and finding out the best possible feature extraction technique. The amino acid composition, dipeptide composition, pseudo amino acid composition and SOCN techniques are implemented. The implementation specification is specified as,

Table 7.6 Implementation specification for hydroxylation PTM

Type	Method/Name
Data Set	PLMD- Hydroxylation (Lysine-K) P=135, N=906, Total= 1041, window size= 15
Feature extraction	AAC, DC, PAAC, SOCN
Dataset balance	Under Sampling(135 each class)
Validation	5-fold cross validation ($2^{-8}$to 2^{8})
Classification	SVM (RBF kernel)
Measuring Matrices	Accuracy, Sensitivity, Specificity

Table 7.7 Parameters of implementation specification for hydroxylation PTM

Methods	AAC	DC	PAAC	SOCN
Parameter	---	---	ƛ= 5	nlag= 5
Size	(1041*20)	(1041*400)	(1041*25)	(1041*10)
C (SVM)	2^{7}	2^{8}	2^{8}	2^{8}
Sigma (RBF)	2^{8}	2^{8}	2^{8}	2^{8}

The table 7.4 shows the specification of implemented system for getting the comparative performance. The hydroxylation dataset is used and lysine, K, is the responsible residue. The imbalance ratio is 6.7. The data balance issue is resolved by using random undersampling method. For random undersampling technique, the majority class is randomly chosen with replacement for n times, here n is the number of observations in the

minority class. Though the undersampling technique discards information to a great extent from the majority class which has adverse effect on the overall performance, the random undersampling technique is chosen for only the comparative performance among the feature extraction techniques.

The table 7.5 contains the information of parameters C and sigma which has been obtained after the complete run of 5-fold cross. The result is obtained by inputting the extracted features into to support vector machine with the specified parameter C and sigma. The tabular representations is shown below,

Table 7.8 Performance comparison among feature extraction technique

Feature extraction techniques	Measuring Matrices	Obtained Results
AAC	Accuracy (%)	72
DC		66
PAAC		70
SOCN		65
AAC	Sensitivity (%)	86
DC		88
PAAC		86
SOCN		54
AAC	Specificity (%)	57
DC		45
PAAC		54
SOCN		56

The tabular representation is not intuitive for comparisons, so the graphical representations is shown below.

From the figure 7.1, it is visible that none of the feature extraction techniques are performance good enough compared to the previously developed tools. So new more efficient feature extraction technique is needed for differentiating the positives and negatives. The new feature extraction technique, peptide sequence evolution based

features, is developed on the basis of existing sequence evolutionary feature extraction technique.

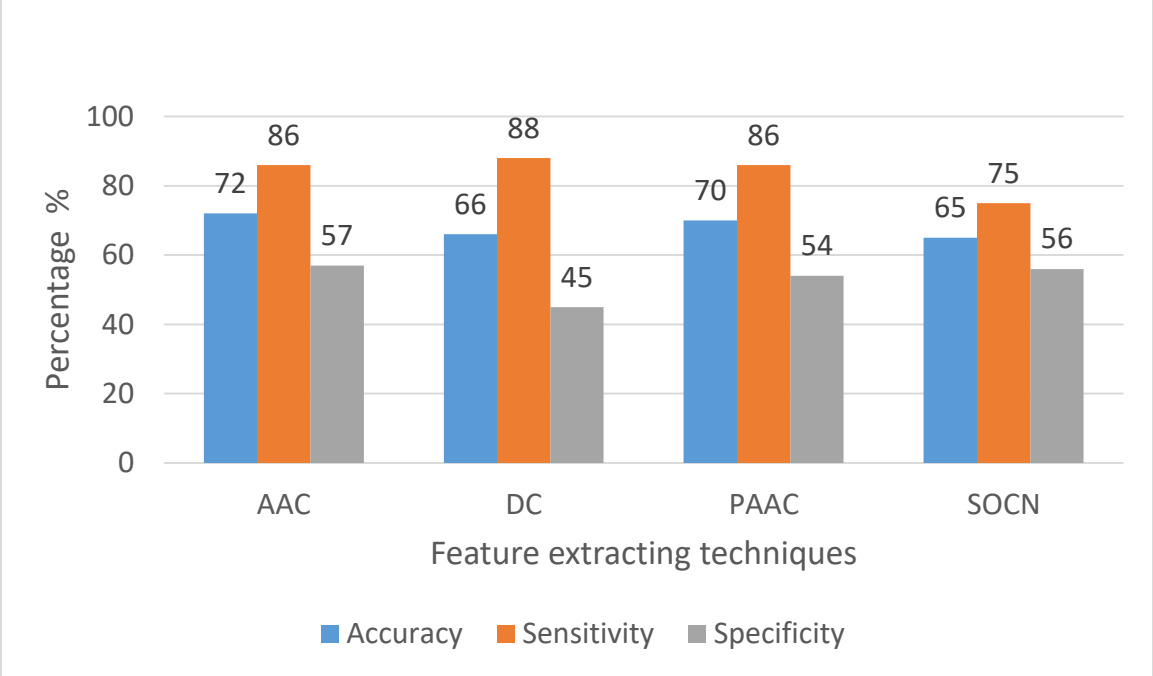


Figure 7.1 Performance comparison among the different feature extraction techniques

The technique is discussed in details in the chapter 4, feature extraction chapter and also in the 7.3 article. This feature extraction technique is implemented for glycation dataset. For calculating the performance, 10-fold cross validation is performed 5 times. The value of C and Sigma has been scanned from 2^{-8} to 2^{8}. For balancing out dataset, oversampling technique has been implemented to get imbalanced ratio of 1:1.

Table 7.9 Optimal C and Sigma value for SVM with RBF kernel

No. of complete run	Lysine (K)	
	C	σ
1st	2^{-8}	2^{1}
2nd	2^{-8}	2^{1}
3rd	2^{-8}	2^{1}
4th	2^{-8}	2^{1}
5th	2^{-8}	2^{1}

The parameters for the PSI-BLAST has also been initialized as follows.

Table 7.10 Parameters for PSI-BLAST

No. of iteration	E-value cutoff
2	0.001

The results are found by calculating the mean and the standard deviation of the 5 results after performing the 10-fold cross validation five times. The result is represented as a mean(± standard deviation).

Table 7.11 Performance comparison between proposed method and previously developed tools

Predictor	Matrices	Lysine (K)
Gly-PseAAC	Acc(%)	68.69(± 0.92)
BPB_GlySite		69.63(± 0.74)
Proposed Predictor		95.94(± 0.54)
Gly-PseAAC	Sn(%)	57.48(± 1.75)
BPB_GlySite		63.68(± 1.40)
Proposed Predictor		98.20(± 0.00)
Gly-PseAAC	Sp(%)	74.30(±1.50)
BPB_GlySite		72.60(± 0.65)
Proposed Predictor		90.67(±1.07)

As the tabular representation is not very intuitive to understand, let's see the graphical representation of the table.

The table 7.11 and figure 9 contains the comparative performance of the previously developed predictors BPB_GlySite and Gly-PseAAC with our proposed method. For comparing the performance among them, Acc, Sn and Sp has been considered. The values of Acc, Sn and Sp are calculated by running 10-fold cross validation 5 times and represented as mean(±standard deviation). The mean and standard deviation is calculated

from the 5 results. The complete execution is run 5 times for the sake of getting a concrete solution.

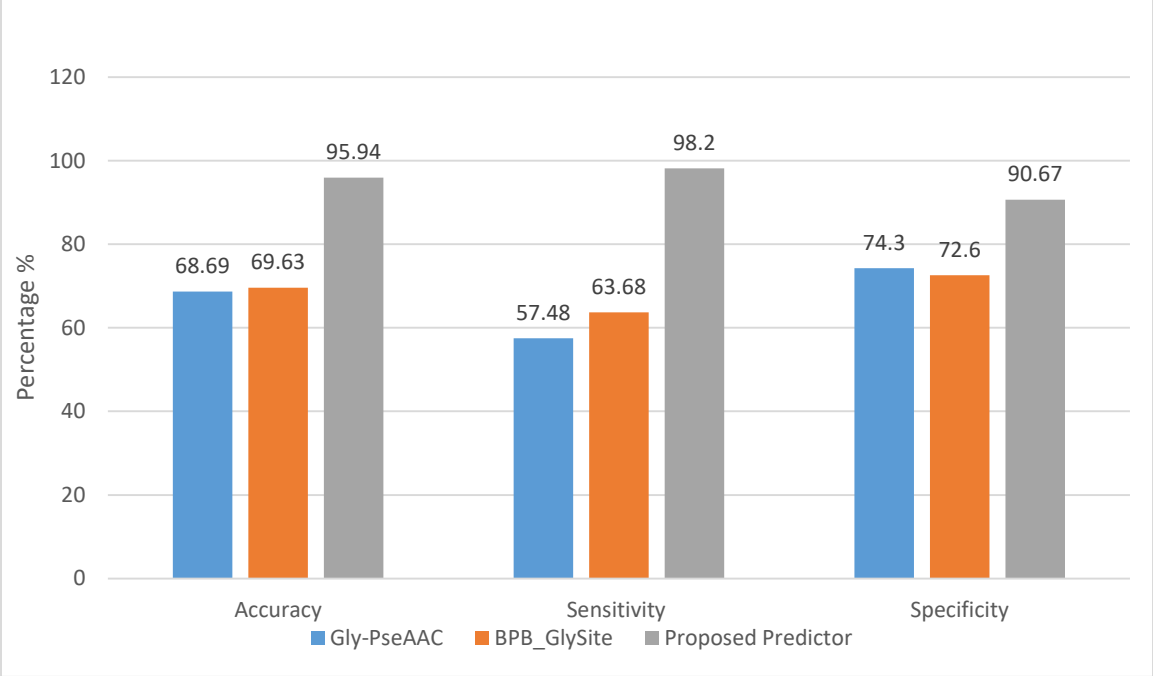


Figure 7.2 Graphical comparison among the previously developed tools and our proposed method

From the table 7.11 and figure 9, it is obvious that in all cases, our proposed model outperforms the existing models BPB_GlySite and Gly-PseAAC to a great extent. PreGly is another predictor which predicts glycated sites with an accuracy of 85.51%. Our proposed method has the accuracy of 95.94(± 0.54)%, which is noticeably higher compared to the PreGly predictor. The first predictor, NetGlycate developed in 2006, provides only information of Matthews correlation coefficient of 0.58, but does not provide any information about Acc, Sn or Sp so that we can compare with our proposed method. The tools developed after the NetGlycate are better in terms of performance. The performance of our proposed method is better compared to the tools developed after NetGlycate. So, a conclusion can be drawn that, our proposed method is better in terms of accuracy, sensitivity and specificity compared to the all previously developed tools NetGlycate, PreGly, Gly-PseAAC and BPB_GlySite.

Conclusion

The BPB_GlySite has used bi-profile bayes feature extraction whereas we have used a new feature extraction technique called peptide sequence evolution based feature. Other than the feature extraction technique, the dataset and the classifier (Support Vector Machine) are same in both of the models. As the performance is better on our proposed method, a conclusion can be drawn that the peptide sequence evolution based feature extraction technique preserves more information than the bi-profile bayes feature extraction technique for separating the positive and negative classes for the dataset of glycation sites.

Chapter 8
Conclusion

8.1 Concluding discussion

The book is the work through of the thesis of building a predictor for efficiently predict the lysine glycation site PTM. For building the model, several steps is required. Each chapter of this book is considered as each step. The data preprocessing chapter consists of several methods of preprocessing. The discussion is kept brief in each chapter because each topic is so vast in reality that it's hard to explain everything in a single book. The relevant methods and topics has been discussed. The relevant means those which are implemented in the implementation and performance analysis, chapter of 7. The main contribution of this book is the introduction of a new feature extraction technique discussed in chapter 4. The new technique named as peptide sequence evaluation based features which is the modification of sequential evolutionary technique that already exists. The difference of those are discussed mathematically and graphically in the chapter 4. The classification is one of the crucial factors in the thesis. From several classification methods, the support vector machine is chosen for ensuring the similarity among the previous works. The previous tools used the support vector machine as classifier. As our contribution is in the feature extraction technique, for focusing the actual performance comparison among the feature extraction techniques the classifier is kept same. For comparing the performances, several measuring techniques are discussed in chapter 6. Some of them has been used for comparing the overall performance of our model. In chapter 7, the performance comparison shows that our feature extraction techniques are performing better than all other predictors, in other words, the new proposed feature extraction technique preserves more information for separating the non-glycated sites to glycated sites compared to all other feature extraction techniques used by the previous predictors.

8.2 Future work

A new feature extraction technique, peptide sequence evaluation based feature extraction, is implemented on lysine glycation dataset. In lysine glycation dataset it is performing better than other predictors, but here we cannot draw the conclusion that this dataset will

perform in all PTMs. To find out whether it is a good generalized predictor, we need to implement on other datasets. If it perform with an acceptable performance we will draw a conclusion about that. Otherwise, we will find the inherent characteristics of dataset for which this method is good for specific datasets. In future, the new feature extraction technique will be implemented on different datasets for finding the generalized performance of this method.

References

[1] Y. Xu, J. Ding, L. Wu and K. Chou, "iSNO-PseAAC: Predict Cysteine S-Nitrosylation Sites in Proteins by Incorporating Position Specific Amino Acid Propensity into Pseudo Amino Acid Composition", *PLoS ONE*, vol. 8, no. 2, p. e55844, 2013.

[2] C. Walsh, S. Garneau-Tsodikova and G. Gatto, "Protein Posttranslational Modifications: The Chemistry of Proteome Diversifications", *Angewandte Chemie International Edition*, vol. 44, no. 45, pp. 7342-7372, 2005.

[3] E. Witze, W. Old, K. Resing and N. Ahn, "Mapping protein post-translational modifications with mass spectrometry", *Nature Methods*, vol. 4, no. 10, pp. 798-806, 2007.

[4] "Amino acid", En.wikipedia.org, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Amino_acid. [Accessed: 28- Oct- 2018].

[5] F. CRICK, "Central Dogma of Molecular Biology", *Nature*, vol. 227, no. 5258, pp. 561-563, 1970.

[6] "Central dogma of molecular biology", En.wikipedia.org, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Central_dogma_of_molecular_biology. [Accessed: 28- Oct- 2018].

[7] Hausman RE, Cooper GM (2004). The cell: a molecular approach.

[8] N. Ahmed, R. Babaei-Jadidi, S. Howell, P. Beisswenger and P. Thornalley, "Degradation products of proteins damaged by glycation, oxidation and nitration in clinical type 1 diabetes", *Diabetologia*, vol. 48, no. 8, pp. 1590-1603, 2005.

[9] e. Ling X, "Immunohistochemical distribution and subcellular localization of three distinct specific molecular structures of advanced glycation end products in... - PubMed - NCBI", *Ncbi.nlm.nih.gov*, 2018.

[10] S. Agalou, "Profound Mishandling of Protein Glycation Degradation Products in Uremia and Dialysis", *Journal of the American Society of Nephrology*, vol. 16, no. 5, pp. 1471-1485, 2005.

[11] M. Johansen, L. Kiemer and S. Brunak, "Analysis and prediction of mammalian protein glycation", *Glycobiology*, vol. 16, no. 9, pp. 844-853, 2006.

[12] Y. Liu, W. Gu, W. Zhang and J. Wang, "Predict and Analyze Protein Glycation Sites with the mRMR and IFS Methods", *BioMed Research International*, vol. 2015, pp. 1-6, 2015.

[13] M. Hasan, J. Li, S. Ahmad and M. Molla, "predCar-site: Carbonylation sites prediction in proteins using support vector machine with resolving data imbalanced issue", *Analytical Biochemistry*, vol. 525, pp. 107-113, 2017.

[14] M. Hasan, S. Ahmad and M. Molla, "iMulti-HumPhos: a multi-label classifier for identifying human phosphorylated proteins using multiple kernel learning based support vector machines", *Molecular BioSystems*, vol. 13, no. 8, pp. 1608-1618, 2017.

[15] Y. Xu, L. Li, J. Ding, L. Wu, G. Mai and F. Zhou, "Gly-PseAAC: Identifying protein lysine glycation through sequences", *Gene*, vol. 602, pp. 1-7, 2017.

[16] Z. Ju, J. Sun, Y. Li and L. Wang, "Predicting lysine glycation sites using bi-profile bayes feature extraction", *Computational Biology and Chemistry*, vol. 71, pp. 98-103, 2017.

[17] A. Meinhart and P. Cramer, "Recognition of RNA polymerase II carboxy-terminal domain by 3′-RNA-processing factors", *Nature*, vol. 430, no. 6996, pp. 223-226, 2004.

[18] "C-terminus", En.wikipedia.org, 2018. [Online]. Available: https://en.wikipedia.org/wiki/C-terminus. [Accessed: 29- Oct- 2018].

[19] K. Starheim, K. Gevaert and T. Arnesen, "Protein N-terminal acetyltransferases: when the start matters", *Trends in Biochemical Sciences*, vol. 37, no. 4, pp. 152-161, 2012.

[20] "Acetylation", En.wikipedia.org, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Acetylation. [Accessed: 29- Oct- 2018].

[21] I. Dalle-Donne, D. Giustarini, R. Colombo, R. Rossi and A. Milzani, "Protein carbonylation in human diseases", *Trends in Molecular Medicine*, vol. 9, no. 4, pp. 169-176, 2003.

[22] I. Møller, A. Rogowska-Wrzesinska and R. Rao, "Protein carbonylation and metal-catalyzed protein oxidation in a cellular perspective", *Journal of Proteomics*, vol. 74, no. 11, pp. 2228-2242, 2011.

[23] D. Bota, H. Van Remmen and K. Davies, "Modulation of Lon protease activity and aconitase turnover during aging and oxidative stress", *FEBS Letters*, vol. 532, no. 1-2, pp. 103-106, 2002.

[24] B. Frohnert, A. Sinaiko, F. Serrot, R. Foncea, A. Moran, S. Ikramuddin, U. Choudry and D. Bernlohr, "Increased Adipose Protein Carbonylation in Human Obesity", *Obesity*, vol. 19, no. 9, pp. 1735-1741, 2011.

[25] I. Dalle-Donne, G. Aldini, M. Carini, R. Colombo, R. Rossi and A. Milzani, "Protein carbonylation, cellular dysfunction, and disease progression", *Journal of Cellular and Molecular Medicine*, vol. 10, no. 2, pp. 389-406, 2006.

[26] W. Qiu, Q. Zheng, B. Sun and X. Xiao, "Multi-iPPseEvo: A Multi-label Classifier for Identifying Human Phosphorylated Proteins by Incorporating Evolutionary Information into Chou′s General PseAAC via Grey System Theory", *Molecular Informatics*, vol. 36, no. 3, p. 1600085, 2016.

[27] "Post-translational modification", En.wikipedia.org, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Post-translational_modification. [Accessed: 05- Nov- 2018].

[28] J. Jia, Z. Liu, X. Xiao, B. Liu and K. Chou, "iCar-PseCp: identify carbonylation sites in proteins by Monte Carlo sampling and incorporating sequence coupled effects into general PseAAC", *Oncotarget*, vol. 7, no. 23, 2016.

[29] "A Review on Imbalanced Data Handling Using Undersampling and Oversampling Technique", *International Journal of Recent Trends in Engineering and Research*, vol. 3, no. 4, pp. 444-449, 2017.

[30] S. Kotsiantis and P. Pintelas, "Mixture of Expert Agents for Handling Imbalanced Data Sets", *Semanticscholar.org*, 2018.

[31] N. Chawla, K. Bowyer, L. Hall and W. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique", *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002.

[32] M. Santos, P. Abreu, P. García-Laencina, A. Simão and A. Carvalho, "A new cluster-based oversampling method for improving survival prediction of

hepatocellular carcinoma patients", *Journal of Biomedical Informatics*, vol. 58, pp. 49-59, 2015.

[33] L. Kuncheva, M. Skurichina and R. Duin, "An experimental study on diversity for bagging and boosting with linear classifiers", *Information Fusion*, vol. 3, no. 4, pp. 245-258, 2002.

[34] B. Sumana and T. Santhanam, "Optimizing the Prediction of Bagging and Boosting", *Indian Journal of Science and Technology*, vol. 8, no. 35, 2015.

[35] "AMINO ACID COMPOSITION OF ANIMAL PROTEIN", Nutrition Reviews, vol. 1, no. 12, pp. 369-370, 1943.

[36] M. Smith, "The amino acid composition of proteins", 2018.

[37] "Correlation between protein stability and its dipeptide composition: Studies on their structural and biological implications", *Protein Engineering*, Design and Selection, 1993.

[38] M. Hayat and A. Khan, "Prediction of membrane protein types by using dipeptide and pseudo amino acid composition-based composite features", *IET Communications*, vol. 6, no. 18, pp. 3257-3264, 2012.

[39] L. Yang, H. Gao, Z. Liu and L. Tang, "Identification of phage virion proteins by using the g-gap tripeptide composition", *Letters in Organic Chemistry*, vol. 15, 2018.

[40] K. Chou, "Prediction of Protein Subcellular Locations by Incorporating Quasi-Sequence-Order Effect", *Biochemical and Biophysical Research Communications*, vol. 278, no. 2, pp. 477-483, 2000.

[41] J. Shi, S. Zhang, Q. Pan and G. Zhou, "Using pseudo amino acid composition to predict protein subcellular location: approached with amino acid composition distribution", *Amino Acids*, vol. 35, no. 2, pp. 321-327, 2008.

[42] K. Chou and Y. Cai, "Predicting protein quaternary structure by pseudo amino acid composition", *Proteins: Structure, Function, and Bioinformatics*, vol. 55, no. 3, pp. 781-781, 2004.

[43] "protr: R package for generating various numerical representation schemes of protein sequences", *Cran.r-project.org*, 2018. [Online]. Available: https://cran.r-project.org/web/packages/protr/vignettes/protr.html#sequence-order-coupling-number. [Accessed: 05- Nov- 2018].

[44] K. Chou, Z. Wu and X. Xiao, "iLoc-Euk: A Multi-Label Classifier for Predicting the Subcellular Localization of Singleplex and Multiplex Eukaryotic Proteins", *PLoS ONE*, vol. 6, no. 3, p. e18258, 2011.

[45] K. Chou, "Structural Bioinformatics and its Impact to Biomedical Science", *Current Medicinal Chemistry*, vol. 11, no. 16, pp. 2105-2134, 2004.

[46] A. Schaffer, "Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements", *Nucleic Acids Research*, vol. 29, no. 14, pp. 2994-3005, 2001.

[47] "Chapter 2 : SVM (Support Vector Machine) — Theory – Machine Learning 101 – Medium", *Medium*, 2018. [Online]. Available: https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72. [Accessed: 05- Nov- 2018].

[48] V. Vapnik, The nature of statistical learning theory. New York: Springer, 2010.

[49] "Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond", *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 781-781, 2005.

[50] M. Hasan, M. Nasser, B. Pal and S. Ahmad, "Support Vector Machine and Random Forest Modeling for Intrusion Detection System (IDS)", *Journal of Intelligent Learning Systems and Applications*, vol. 06, no. 01, pp. 45-52, 2014.

[51] A. KOWALCZYK, "Linear Kernel: Why is it recommended for text classification ?", *SVM Tutorial*, 2018. [Online]. Available: https://www.svm-tutorial.com/2014/10/svm-linear-kernel-good-text-classification/. [Accessed: 05- Nov- 2018].

[52] J. Jia, Z. Liu, X. Xiao, B. Liu and K. Chou, "pSuc-Lys: Predict lysine succinylation sites in proteins with PseAAC and ensemble random forest approach", *Journal of Theoretical Biology*, vol. 394, pp. 223-230, 2016.

[53] Z. Ju, J. Cao and H. Gu, "Predicting lysine phosphoglycerylation with fuzzy SVM by incorporating k-spaced amino acid pairs into Chouʹs general PseAAC", *Journal of Theoretical Biology*, vol. 397, pp. 145-150, 2016.

[54] H. Lv, J. Han, J. Liu, J. Zheng, R. Liu and D. Zhong, "CarSPred: A Computational Tool for Predicting Carbonylation Sites of Human Proteins", *PLoS ONE*, vol. 9, no. 10, p. e111478, 2014.

[55] Y. Xu, Y. Ding, N. Deng and L. Liu, "Prediction of sumoylation sites in proteins using linear discriminant analysis", *Gene*, vol. 576, no. 1, pp. 99-104, 2016.

[56] B. Liu, Y. Liu, X. Jin, X. Wang and B. Liu, "iRSpot-DACC: a computational predictor for recombination hot/cold spots identification based on dinucleotide-based auto-cross covariance", *Scientific Reports*, vol. 6, no. 1, 2016.

[57] Z. Liao, Y. Ju and Q. Zou, "Prediction of G Protein-Coupled Receptors with SVM-Prot Features and Random Forest", *Scientifica*, vol. 2016, pp. 1-10, 2016.

[58] "Sensitivity and specificity", *En.wikipedia.org*, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Sensitivity_and_specificity. [Accessed: 05- Nov- 2018].

[59] "Receiver operating characteristic", *En.wikipedia.org*, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Receiver_operating_characteristic. [Accessed: 05- Nov- 2018].