PREDICTION OF PERMISSIVE INSERTION SITES IN PROTEINS

by

Husamaldin Tayeh

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of the requirements for the degree of
Master of Science

Sabancı University

January 2013

PREDICTION OF PERMISSIVE INSERTION SITES IN PROTEINS

APPROVED BY:

Prof. Dr. Osman Uğur Sezerman
(Thesis Advisor) ....

Assoc. Prof. Erkay Savaş ....

Asst. Prof. Hüsnü Yenigün ....

Asst. Prof. Kemal Kiliç ....

Assoc. Prof. Yücel Saygın ....

DATE OF APPROVAL: ....

PREDICTION OF PERMISSIVE INSERTION SITES IN PROTEINS

Husamaldin Tayeh

CS, Master's Thesis, 2013

Thesis Supervisor: Prof. Dr. Osman Uğur Sezerman

Keywords: Permissive Insertion Sites, Dipeptide Composition, SVM, Feature Selection

Abstract

The procedure of domain insertion is proven to be very effective in the process of creating modified proteins that can be used for different protein engineering applications. Domain insertion alters the functionality of the protein by inserting gene or genes into certain domains. Proteins usually tolerate insertions in specific sites only, therefore identifying those permissive insertion sites is crucial for any successful insertion attempt. Normally, determining permissive insertion sites is performed experimentally by a genetic approach. However an educated guess can assist in predicting the potential permissive insertion sites.

In this work, we introduced a method for predicting permissive insertion sites through the utilization of machine learning and data mining techniques. We have adopted an educated guess approach to predict permissive sites by extracting distinctive features from the amino acids surrounding the insertion site included within any captured amino acid window. The window size was made adjustable and can capture any odd number of amino acids. We used a number of features related to amino acids obtained from this window and then used a machine learning based approach to construct a trained SVM model using 135 permissive and non-permissive sites obtained from 10 different proteins.

Our trained model was used to predict permissive insertion sites in Outer membrane usher protein FasD, Lactose operon repressor LacI, Type II secretion system protein XpsD, and Maltose periplasmic protein MalE and 70.59%, 61.11%, 61.90% and 90.00% accuracies were achieved respectively.

PROTEİNLERDE İZİN VERİLEN GEN YERLEŞTİRME ALANLARININ TAHMİN EDİLMESİ

Husamaldin Tayeh

CS, Master Tezi, 2013

Tez Danismani: Prof. Dr. Osman Uğur Sezerman

Anahtar Kelimeler: Gen Yerleştirmeye Elverişli Protein Alanları, Dipeptit Kompoziyon, SVM, Özellik Seçimi

Özet

Proteinlere farklı proteinlerin eklenmesi yönteminin, birbirinden farklı protein mühendisliği uygulamalarında kullanılan farklılaştırılmış protein üretimi sürecindeki etkinliği kanıtlanmıştır. Protein ekleme, proteini ifade eden gen üstünde belli başlı bölgelere gen yerleştirerek farklı bir protein elde edilir ve proteinin işlevselliğinde değişikliğe yol açar. Proteinler, sadece üzerlerinde belli başlı alanlara yapılan yerleştirmeleri tolere edebilirler. Bu yüzden bu tolere edilen yerleştirme alanlarının tanımlanması, başarılı bir yerleştirme yapabilmek için büyük önem taşır. Bu yerleştirime alanları, deneme yanılma yöntemiyle tanımlanabilir. Fakat, bu alanlara yönelik doğruluk oranı yüksek bir tahmin yöteminin geliştirilmesi, bu alanların ortaya çıkarılmasını kolaylaştıracaktır.

Bu çalışmada, makina öğrenmesi ve veri madenciliği yöntemlerini kullanarak, proteinlerin tolere edilebilen gen yerleştirme alanlarını tahmin etmekteyiz. Bu tahminler eğitilmiş tahminler olarak adlandırmaktayız. Eğitilmiş tahminlere, gen yerleştirme alanını çevreleyen amino asitlerin belirgin özelliklerini seçerek ulaşmaktayız. Bu tek sayıdaki yerleştirme bölgesini çevreleyen amino asitleri, boyu ayarlanabilen bir pencere yardımıyla belirlemekteyiz. Yerleştirme bölgesi, bu pencerenin ya merkez noktasına ya da orta değer noktasına düşmektedir. Bu pencere içerisinden, amino asitlerle alakalı bir grup özellik elde edilmiştir. Sonrasında, SVM makine öğrenme yöntemini kullanılarak, 10 farklı proteinden elde edilen gen yerleştirmeye elverişli ve elverişsiz 135 bölge ile eğitilerek bir model oluşturulmuştur.

Eğitilmiş modelimiz, Dış zar yer gösterici proteini FasD, Laktoz kalıt baskılayıcı LacI, Tip II sekresyon sistemi proteini XpsD ve de Maltoz periplazmik proteini MalE için sırasıyla %70.59, %61.11, %61.90 ve %90.00 doğruluk oranlarına erişmiştir.

"To my parents

&

my wife Iman"

Acknowledgements

It is a great pleasure to extend my gratitude to my thesis advisor Prof. Dr. Osman Uğur Sezerman for his precious guidance and support. I am greatly indebted to him for his supervision and excellent advises throughout my Master study.

I would gratefully thank Assoc. Prof. Erkay Savaş, Asst. Prof. Hüsnü Yenigün, Asst. Prof. Kemal Kiliç and Assoc. Prof. Yücel Saygın for spending their valuable time to serve as my jurors.

I would like to acknowledge the financial support provided by Erasmus Mundus LOT3 project.

Finally, I would like to thank my family and my wife for all their love and support throughout my life.

Contents

1 INTRODUCTION 1

1.1 Motivation 1

1.2 Organisation of Thesis 2

2 BACKGROUND AND RELATED WORK 3

2.1 Biological Background 3

2.1.1 Protein 3

2.1.2 Peptide Bonds 5

2.1.3 Protein Secondary Structure 5

2.1.4 Permissive Sites 6

2.1.5 Domain 8

2.1.6 Types of Domain Insertions 8

2.1.7 Characteristics of Glycine 9

2.2 Support Vector Machine 9

2.3 Feature Selection 11

2.4 Related work 13

3 METHODOLOGY 15

3.1 Introduction 15

3.2 Feature Extraction 17

3.2.1 Calculation of Dipeptide Composition 18

3.2.2 Polarity 19

3.2.3 Charge 20

3.2.4 Hydropathy and Hydrophilicity 21

3.2.5 Corresponding Secondary Structure Sequence 21

3.2.6 Amino Acids Occurrences 22

3.2.7 Presence of Glycine 23

3.2.8 Feature Vector 23

3.3 Dataset Collection 24

3.4 Classifier Training 24

4 EXPERIMENTS AND RESULTS 26

4.1 Dataset 26

4.2 Training and Validation 26

4.3 Experiments with 5-Fold Cross-Validation 29

4.4 Experiments with Feature Selection 30

4.4.1 Wrapper Method 32

4.4.2 Filter Method 32

4.5 Results 32

5 CONCLUSION AND FUTURE WORKS 39

List of Figures

2.1 Illustration of the three groups that all amino acids contain. The R side chain differs with each amino acid and determines the properties of the amino acid [29] 5

2.2 Peptide bond formation by dehydration synthesis. The amino acids glycine and alanine combine to form a dipeptide. The newly formed bond between the carbon atom of glycine and the nitrogen atom of alanine is called a peptide bond [38] . . . 6

2.3 Secondary structure: helix and pleated sheets (with three polypeptide strands) [38] 7

2.4 Schematic representation of types of domain insertions observed in protein structures. (a) Single insertion. (b) Nested insertion. insert1 N' and insert1 C' represent the N and C terminus of insert, respectively. (c) Two-domain insertion. (d) Three-domain insertion [2] 9

2.5 Structural formula of the Glycine [38] 10

2.6 Classification (linear seperabale case) [12] 11

3.1 A schematic illustration of the methodology 16

3.2 An illustrationof the first position scenario. (a) First position problem where features extraction starts at position 5. (b) Our proposed solution 17

3.3 Illustration of a window capturing 9 amino acids between the positions 51 and 59 inclusive. The amino acid at the position 55 represents the insertion site being examined 18

3.4 Construction of k-letter words where $k = 2$. (a) Resulting k-letter words. (a) Representation by group number 19

3.5 Sample residues of the XpsD protein and their corresponding secondary structure 22
4.1 Train and test [7] 27
4.2 5-fold cross-validation 29

List of Tables

2.1 List of amino acids 4

3.1 Amino acid groups 19

3.2 Numerical representation for possible amino acid side-chain polarity 20

3.3 Numerical representation for possible amino acid side-chain charge 21

3.4 Numerical representation for possible secondary structure shapes 23

3.5 List of all extracted features and their possible numerical representations 24

3.6 Possible number of features generated for $w = 7$ 25

4.1 Training dataset 28

4.2 Test dataset 28

4.3 Training set results for different window sizes using SVM classifier with 5-fold cross-validation 30

4.4 Test set results for different window sizes using SVM trained model 31

4.5 Confusion matrices of test set results for different window sizes using SVM trained model 31

4.6 Training set results for different window sizes using SVM classifier with 5-fold cross-validation after applying wrapper method feature selection 33

4.7 Test set results for different window sizes using SVM trained model after applying wrapper method feature selection 34

4.8 Training set results for different window sizes using SVM classifier with 5-fold cross-validation after applying filter method feature selection 35

4.9 Test set results for different window sizes using SVM trained model after applying filter method feature selection 36

List of Abbreviations

D	Asparatic Acid
DC	Dipeptide Composition
DNA	Dioxyribonucleic Acid
G	Glycine
H	Histidine
LIBSVM	A Library for Support Vector Machines
M	Methionine
RMSE	Root Mean Square Error
RNA	Ribonucleic Acid
SVM	Support Vector Machine
T	Threonine
WEKA	Waikato Environment for Knowledge Analysis

Chapter 1

1 INTRODUCTION

1.1 Motivation

As a result of the Human Genome Project and related efforts, DNA (dioxyribonucleic acid), RNA (ribonucleic acid), and protein data accumulate at an accelerating rate. Mining these biological data to extract useful knowledge is essential in genome processing. This subject has recently gained significant attention in the bioinformatics community [1, 4].

Proteins play a very essential role in the cell which controls and affects all functions. Their role is mainly determined by their structure. Likewise, it is the amino acid sequence that determines the protein's structure. Therefore, there is a strong relationship among the sequence, structure and function of the proteins [32]. Protein modification and engineering hold great significance for the future of medicine and biotechnology. Modification of genes at the nucleotide level continues to provide relevant insights into the structural elements critical to gene and protein function [16]. The procedure of domain insertion is proven to be very effective in the process of creating modified proteins that can be used for different protein engineering applications. Domain insertion alters the functionality of the protein by inserting gene or genes into certain domains. Proteins usually allow insertions in specific sites only, therefore identifying those permissive sites is crucial for any successful domain insertion attempt. Usually, the procedure for determining the permissive insertion sites in a protein is performed in biological laboratories by a method proven to be neither cost nor labor effective. The rapid advancements in the field of molecular biology has consequently increased the demand for more robust computational solutions. Moreover, the availability of various state of the art machine learning algorithms and techniques has encouraged more and more scientists to try and solve problems addressed by the biotechnology industry. Those were the main reasons that motivated us to conduct the experiment discussed in this thesis.

1.2 Organisation of Thesis

The organization of the thesis as follows: Chapter 2 presents a brief biological background and an overview of the related works. In Chapter 3, we explain our approach in detail. Chapter 4 discusses the experiments and the results. Lastly, the conclusions and the future works are given in Chapter 5.

Chapter 2

2 BACKGROUND AND RELATED WORK

2.1 Biological Background

2.1.1 Protein

Proteins are organic molecules that contain carbon, hydrogen, oxygen, and nitrogen. Some also contain sulfur. They weigh more than all other organic compounds found in a living cell. In fact, hundreds of different proteins can be found in any single cell, and together they make up 50% or more of a cell's dry weight [38]. Proteins are composed of numerous combinations of 20 major amino acids joined together by peptide bonds. These amino acids are listed in Table 2.1. The properties of a protein depend mainly on its shape, which in turn depends on the arrangement of the amino acids that make up the protein [19, 29]. Every amino acid contains at least one carboxyl ($-$COOH) group and one amino ($-NH_2$) group attached to the same carbon atom, called an alpha-carbon (written Cα) [38]. This carbon atom is also bonded to a side chain which gives each amino acid its characteristics properties such as hydrophobicity, charge and volume Figure 2.1 [29]. Since these properties affect the interactions of amino acid residues, they have a great influence on protein three-dimensional structure and as a result protein's main function. The distribution of hydrophobic and hydrophilic (polar and charged) amino acids dictates the structure of the protein where the hydrophobic residues try to get away from water and hence take a position inside the protein core while the hydrophilic ones prefer to be outside. When amino acids are joined together into a polypeptide chain, a water molecule is released from each joined amino acids. Therefore, rather than the original amino acids, the protein is composed of amino acid residues [36]. These amino acid residues form what is known as the primary structure of the protein [8].

Amino Acid	Abbreviations		Polarity	Charge	Hypath[1]	Hyphil[2]
Alanine	Ala	A	nonpolar	neutral	1.8	-0.87
Arginine	Arg	R	polar	positivel	-4.5	15.86
Asparagine	Asn	N	polar	neutral	-3.5	7.58
Aspartic acid	Asp	D	polar	negative	-3.5	9.66
Cysteine	Cyc	C	nonpolar	neutral	2.5	-0.34
Glutamic acid	Glu	E	polar	negative	-3.5	7.75
Glutamine	Gln	Q	polar	neutral	-3.5	6.48
Glycine	Gly	G	nonpolar	neutral	-0.4	0
Histidine	His	H	polar	positive	-3.2	5.6
Isoleucine	Ile	I	nonpolar	neutral	4.5	-3.98
Leucine	Leu	L	nonpolar	neutral	3.8	-3.98
Lysine	Lys	K	polar	positive	-3.9	6.49
Methionine	Met	M	nonpolar	neutral	1.9	-1.41
Phenylalanine	Phe	F	nonpolar	neutral	2.8	-2.04
Proline	Pro	P	nonpolar	neutral	-1.6	-0.01
Serine	Ser	S	polar	neutral	-0.8	4.34
Threonine	Thr	T	polar	neutral	-0.7	3.51
Tryptophan	Trp	W	nonpolar	neutral	-0.9	-1.39
Tyrosine	Tyr	Y	polar	neutral	-1.3	1.08
Valine	Val	V	nonpolar	neutral	4.2	-3.1

[1] Hydropathy Index
[2] Hydrophilicity Index

Table 2.1: List of amino acids

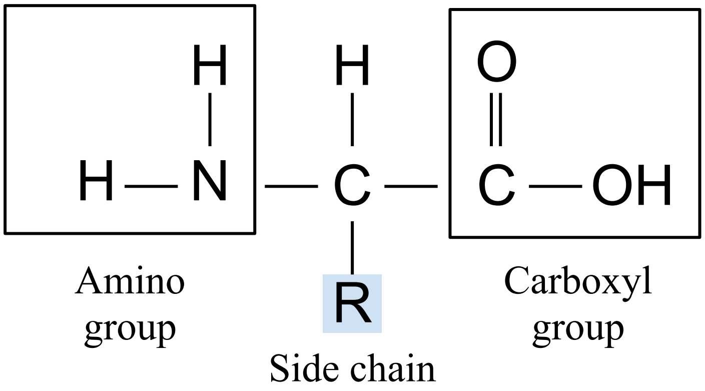


Figure 2.1: Illustration of the three groups that all amino acids contain. The R side chain differs with each amino acid and determines the properties of the amino acid [29]

2.1.2 Peptide Bonds

Amino acids bond between the carbon atom of the carboxyl ($-$COOH) group of one amino acid and the nitrogen atom of the amino($-NH_2$) group of another. The bonds between amino acids are called peptide bonds Figure 2.2. For every peptide bond formed between two amino acids, one water molecule is released; thus, peptide bonds are formed by dehydration synthesis. When two amino acids are joind together by a peptide bond the resulting compound is called dipeptide. Adding another amino acid to a dipeptide would form a tripeptide. Further additions of amino acids would produce a long, chainlike molecule called a peptide (4-9 amino acids) or polypeptide (10-2000 or more amino acids) [38].

2.1.3 Protein Secondary Structure

Proteins have four levels of structure: primary, secondary, tertiary, and quaternary [29]. In biochemistry and structural biology, secondary structure

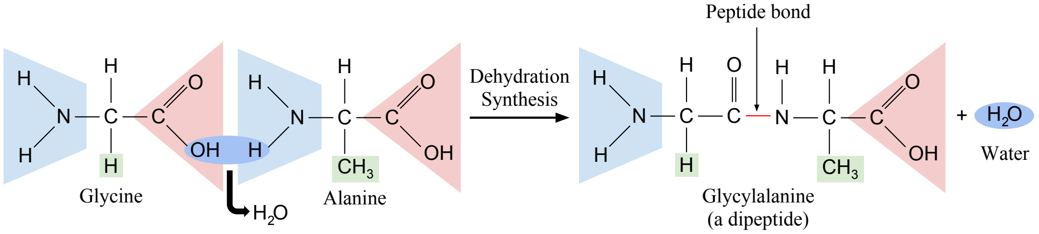


Figure 2.2: Peptide bond formation by dehydration synthesis. The amino acids glycine and alanine combine to form a dipeptide. The newly formed bond between the carbon atom of glycine and the nitrogen atom of alanine is called a peptide bond [38]

is the general three-dimensional form of local segments of biopolymers such as proteins and nucleic acids (DNA/RNA) [38]. A protein's secondary structure is the localized, repetitious twisting or folding of the polypeptide chain. This aspect of a protein's shape results from hydrogen bonds joining the atoms of peptide bonds at different locations along the polypeptide chain. Certain sequences of amino acids will arrange themselves into clockwise spirals or helical structure termed alpha (α) helix while the roughly parallel portions of the chain will form a pleated structure termed beta (β) sheet Figure 2.3 [29]. Both structures are held together by hydrogen bonds between oxygen or nitrogen atoms that are part of the polypeptide's backbone [38].

2.1.4 Permissive Sites

Permissive sites are defined as regions of the protein that are likely to be flexible enough to accommodate inserts without damage to the protein biogenesis, final localization, and folding [42]. Proteins can be remarkably tolerant of major mutational changes. Sites that accommodate large insertions without loss of function (permissive sites) appear generally to correspond to

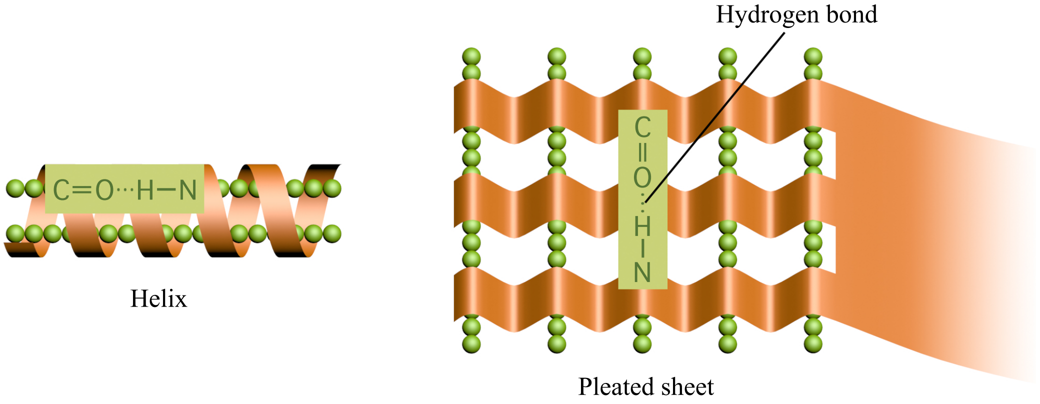


Figure 2.3: Secondary structure: helix and pleated sheets (with three polypeptide strands) [38]

surface regions at which the added sequences do not disrupt overall folding. The identification of such sites can aid in the engineering of functional derivatives of a protein with novel properties [24].

In order to find sites appropriate for insertion and cell surface exposure, one may proceed with an educated guess or with an experimental approach. The more one knows about the protein having similar properties related to their sequence similarity, the more one may expect the educated guess approach to work. However, if nothing is known about the protein and that the gene has been cloned, one may use directly an experimental approach to determine permissive sites. The educated guess approach consists of trying to predict permissive sites by identifying certain sequence and structural features of the experimentally successful permissive sites. For example, flexible regions of the protein and among them try to select those that are likely to be cell surface exposed. Hydrophilic sequences or regions predicted as turns can also be good candidates for permissive sites [42].

2.1.5 Domain

Domains are distinct functional and/or structural units in a protein. Usually they are responsible for a particular function or interaction, contributing to the overall role of a protein. Domains may exist in a variety of biological contexts, where similar domains can be found in proteins with different functions. Proteins can comprise a single domain or a combination of domains hence called multi-domain [2]. Therefore, domains are very important in finding protein's function, classifying protein's fold, and identifying homology relationships. In multi-domain proteins, each domain can have a different function independent from the others, or they can work together in a concerted action. Domains form the functionally important sites of the proteins such as the catalytic sites of the enzymes or ligand binding sites. Moreover, since domains can fold independently, they play a significant role in protein folding by accelerating the folding process and reducing the potentially large combination of residue interactions.

2.1.6 Types of Domain Insertions

Domain insertions can be categorized as either single or multiple depending on the number of inserts as shown in Figure 2.4. In single insertions, one domain is inserted into another domain, and both domains can belong to the same or different superfamilies. In multiple insertions, more than one domain, either of the same or different superfamily, is inserted into the parent domain [2]. There are three types of multiple insertions:

1. Nested insertions

2. Two-domain insertions

3. Three-domain insertions

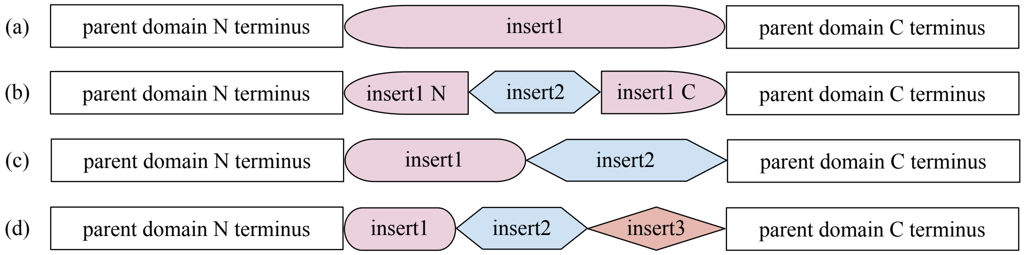


Figure 2.4: Schematic representation of types of domain insertions observed in protein structures. (a) Single insertion. (b) Nested insertion. insert1 N' and insert1 C' represent the N and C terminus of insert, respectively. (c) Two-domain insertion. (d) Three-domain insertion [2]

2.1.7 Characteristics of Glycine

Glycine (abbreviated as Gly or G) is an organic compound with the formula NH_2CH_2COOH. Having a hydrogen substituent as its side-chain, glycine is the smallest of the 20 amino acids commonly found in proteins. Its codons are GGU, GGC, GGA, GGG. the genetic code. Glycine is a colourless, sweet-tasting crystalline solid. It is unique among the proteinogenic amino acids in that it is not chiral. Glycine is considered the most flexible among the amino acids for its ability to fit into any configuration hydrophilic or hydrophobic environments, and that is due to its minimal side chain of only one hydrogen atom [26]. The structural formula of the Glycine is illustrated in Figure 2.5.

2.2 Support Vector Machine

The Support Vector Machine (SVM) is a state-of-the-art classification method introduced in 1992 by Boser, Guyon, and Vapnik [6]. The SVM classifier is

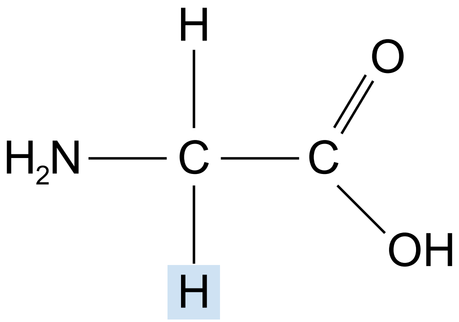


Figure 2.5: Structural formula of the Glycine [38]

widely used in bioinformatics (and other disciplines) due to its high accuracy, ability to deal with high-dimensional data such as gene expression, and flexibility in modeling diverse sources of data [35]. SVMs belong to the general category of kernel methods. A kernel method is an algorithm that depends on the data only through dot-products. When this is the case, the dot product can be replaced by a kernel function which computes a dot product in some possibly high dimensional feature space. This has two advantages: First, the ability to generate non-linear decision boundaries using methods designed for linear classifiers. Second, the use of kernel functions allows the user to apply a classifier to data that have no obvious fixed-dimensional vector space representation. The prime example of such data in bioinformatics are sequence, either DNA or protein, and protein structure. SVM performs classification by constructing an N-dimensional hyperplane that optimally separates the data into two categories. The basic aim is to classify the items that are similar in their feature values. These supervised learning algorithms are known to be enhanced from linear classifiers. The input victors are mapped to a higher dimensional space and data is separated with a hyperplane. Another two extra hyperplanes parallelly positioned on both

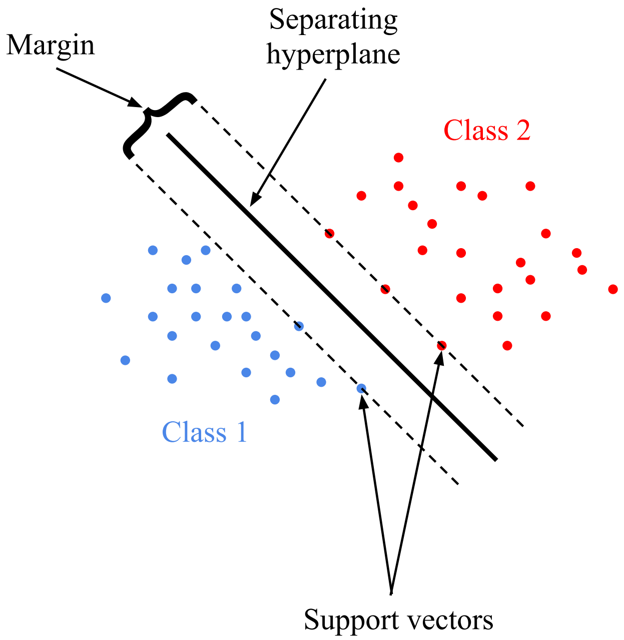


Figure 2.6: Classification (linear seperabale case) [12]

sides of the hyperplane are also invented. The generalization is known to be better as the margin between two parallel hyperplanes is larger. Thus, the distance between these two parallel hyperplanes, is aimed to be maximized, while the effects of the classification error is minimized [10] (Figure 2.6).

2.3 Feature Selection

Feature selection is frequently used as a preprocessing step to machine learning. It is a process of choosing a subset of original features so that the feature space is optimally reduced according to a certain evaluation criterion. Feature selection has proven to be effective in removing irrelevant and redundant features, increasing efficiency in learning tasks, improving learning performance like predictive accuracy, and enhancing comprehensibility of learned results [5]. The large number of features may cause serious problems to many machine learning algorithms with respect to scalability and learning performance. For example, high dimensional data (i.e., data sets with hundreds or thousands of features) can contain high degree of irrelevant and redundant information which may greatly degrade the performance of learning algorithms. Therefore, feature selection becomes very necessary for machine learning tasks when facing high dimensional data. Given an input feature vector $X = \{x_1, x_2, ..., x_M\}$, then the output Y is not determined by the complete set of the input features, instead, it is decided only by a subset of them, such that $X = \{x_{(1)}, x_{(2)}, ..., x_{(m)}\}$, where $m < M$. With sufficient data and time, it is acceptable to use all the input features, including those irrelevant features, to approximate the underlying function between the input and the output. But in practice, there are two problems which may be evoked by the irrelevant features involved in the learning process.

1. The irrelevant input features will induce greater computational cost.

2. The irrelevant input features may lead to overfitting.

Feature selection algorithms fall into two broad categories, the filter model or the wrapper model [13,18]. The filter model relies on general characteristics of the training data to select some features without involving any learning algorithm. The wrapper model requires one predetermined learning algorithm in feature selection and uses its performance to evaluate and determine which features are selected. As for each new subset of features, the wrapper model needs to learn a hypothesis (or a classifier). It tends to find features better suited to the predetermined learning algorithm resulting in superior learning performance, but it also tends to be more computationally expensive than the filter model [20]. When the number of features becomes very large, the filter model is usually chosen due to its computational efficiency.

2.4 Related work

There is a decent number of molecular experiments that were aimed at identifying permissive insertion sites in various proteins. Those experiments formed our main source for constructing the dataset used in this thesis. One related work in the field is the publication of Manoil and Bailey, which is about identifying permissive insertion sites of the *Escherichia coli* lac permease. Their research resulted in identifying 11 permissive insertion sites out of the 20 insertions performed on the *E. Coli* [24].

Unlike the genetic approach, the literature seems to be really poor with computational experiments related to our work. We have however, found one attempt to predict permissive insertion sites by calculating the ’permissibly score’ of amino acid ’windows’ [39]. The aforementioned work inspired us to use a rolling window as an effective method to extract distinctive features. The idea of using a window of amino acid residues was also mentioned by Ofrana and Rosta in a publication titled "Predicted protein-protein interaction sites from local sequence information" [30].

The literature shows that a number of experiments on various protein classification problems were based on the calculation of Dipeptide Composition (DC). The work of Pasquale Petrilli in the publication titled "Classification of protein sequences by their dipeptide composition" formed a fundamental cornerstone in our approach [31].

As for the machine learning techniques, the literature is really rich with publications about various classification problems, classification and prediction techniques. Many of these publications were dealing with proteins and amino acids which made them even more relevant to our work. In fact, as a result of reviewing some of the bioinformatics publications dealing with classification problems we decided to use SVM as the base classifier in our work.

Chapter 3

3 METHODOLOGY

3.1 Introduction

In our approach, we used a moving window with adjustable size to capture amino acids surrounding the insertion site being examined. The insertion site of interest is always at a position equal to the median value of the residues included by any given window. It is also worth mentioning that the window size w was always set to be an odd number in order to be able to extract equal number of features from the other amino acids surrounding the amino acid of interest. Window size can be any odd number that is greater than 1 and less than or equals the sequence length. Each insertion site is represented by a set of features extracted from the other amino acids included within the window. Figure 3.3 illustrates the idea of the moving window. The schematic illustration in Figure 3.1 gives a general overview of our methodology.

Our system takes a protein sequence represented by amino acids and its corresponding secondary structure as inputs. After the feature extraction process is complete our system, uses a trained SVM model to predict the permissive insertion sites.

During the development stage of our methodology we tried to find a solution to how to extract features for the first and last position in the sequence. For example a $w = 9$ can not process the first 4 amino acids, where

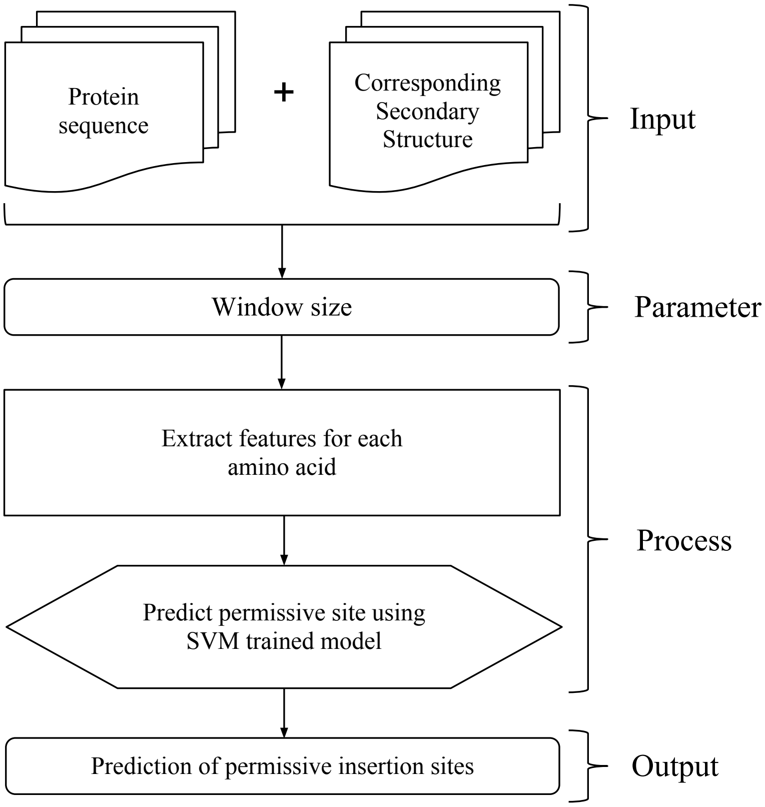

Protein sequence
+
Corresponding Secondary Structure
Input
Window size
Parameter
Extract features for each amino acid
Predict permissive site using SVM trained model
Process
Prediction of permissive insertion sites
Output

Figure 3.1: A schematic illustration of the methodology

the middle amino acid in the window is at the fifth position. To overcome this issue we decided to add zeros to the beginning and end of the sequence. The appropriate number of zeros z is given by Equation 1.

$$z = \frac{w-1}{2} \tag{1}$$

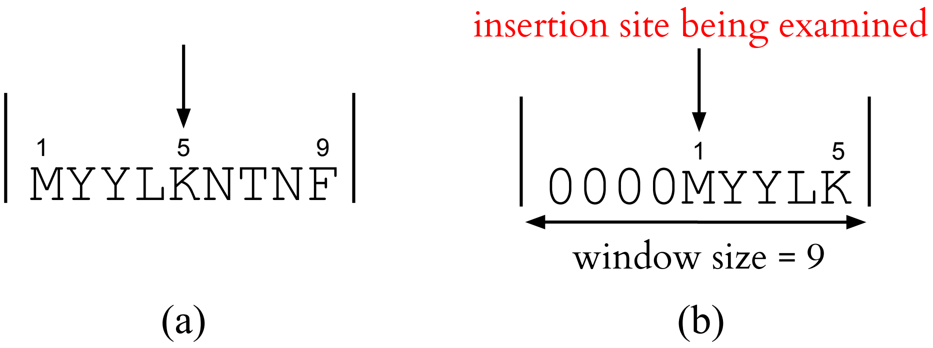


Figure 3.2: An illustrationof the first position scenario. (a) First position problem where features extraction starts at position 5. (b) Our proposed solution

3.2 Feature Extraction

From a one-dimensional point of view, a protein sequence contains characters from the 20-letter amino acid alphabet $\mathcal{A} = \{$A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y$\}$. An important issue in applying any classifier to protein sequence classification is how to encode protein sequences, i.e., how to represent the protein sequences as the input of the classifier. Indeed, sequences may not be the best representation at all. Good input representations make it easier for the classifier to recognize underlying regularities. Thus, good input representations are crucial to the success of classifier learning [41].

Different distinctive sets of features were extracted and then joined together to represent each permissive or non-permissive site. The following sections explain how each group of features was extracted.

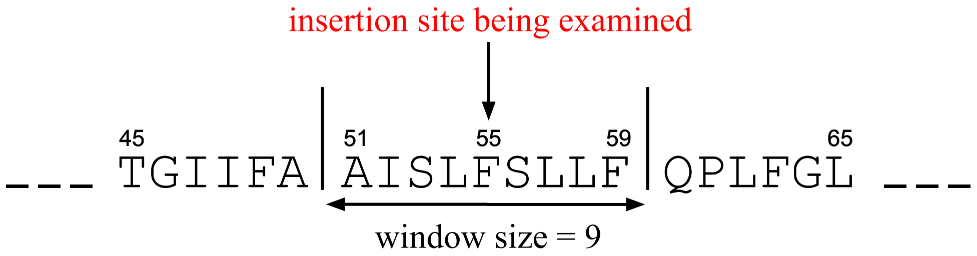

insertion site being examined
45
51
55
59
65
___ TGIIFA AISLFSLLF QPLFGL ___
window size = 9

Figure 3.3: Illustration of a window capturing 9 amino acids between the positions 51 and 59 inclusive. The amino acid at the position 55 represents the insertion site being examined

3.2.1 Calculation of Dipeptide Composition

Normally, The Dipeptide Composition (DC) gives 400 features, defined as:

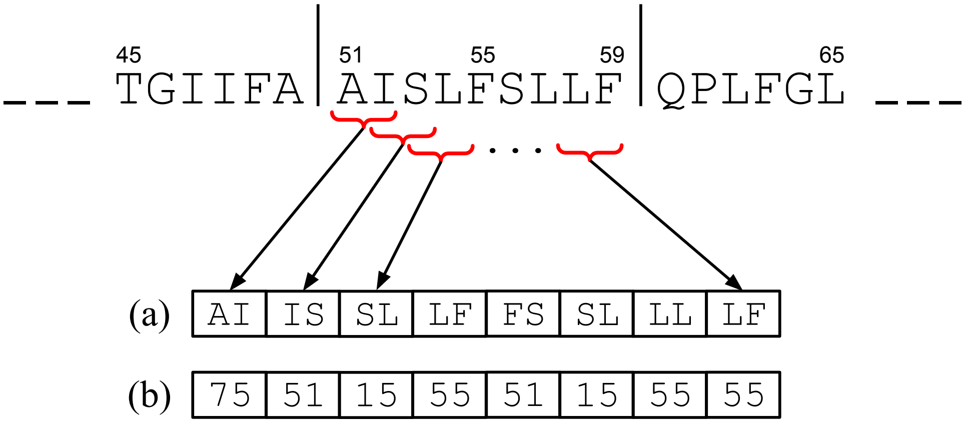
$$f(r,s) = \frac{N_{rs}}{N-1} \qquad r,s = 1,2,...,20. \tag{2}$$

where N_{rs} is the number of dipeptide represented by amino acid type r and type s. However, in our method we have divided the amino acids into 9 different groups according to their side chain properties so the range of possible values of r and s in equation 2 is changed to $r,s = 1,2,...,9$ which, leaves us with $9 \times 9 = 81$ features. Amino acid groups are listed in Table 3.1.

Referring to the window shown in Figure 3.4 we can replace each amino acid by its corresponding group number then list the dipeptide possibilities before calculating the fraction. A window of size w will produce $w-1$ pairs of dipeptide bonds. The resulting pairs of amino acids are sometimes referred to as k-letter word where in this case $k=2$. The occurrence of each pair of amino acids was calculated and turned into a fraction. Our dipeptide calculation section of the program outputs the fraction of each corresponding pair.

Group no.	Amino Acid
1	N, Q, S, T, Y
2	H, K, R
3	E ,D
4	G
5	F, I, L, M, V
6	C
7	A
8	P
9	W

Table 3.1: Amino acid groups



(a)	AI	IS	SL	LF	FS	SL	LL	LF
(b)	75	51	15	55	51	15	55	55

Figure 3.4: Construction of k-letter words where $k = 2$. (a) Resulting k-letter words. (a) Representation by group number

3.2.2 Polarity

Amino acids can be divided into two groups based on their side-chain polarity. They can be either polar or non-polar. The number of features extracted in this step is equal to the window size w. Moreover we have added an extra feature called window polarity to represent the overall polarity of the window. Window polarity p is determined by summing the polarity of each residue included by the window then if the sum is greater than zero, the window polarity is set to '1' and '0' otherwise as shown in 3. The numerical representations for all possible amino acid side-chain polarity are shown in Table 3.2.

$$p = \begin{cases} 1, & \text{if sum of residues polarity} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Side-chain polarity	Numerical representation
Polar	1
Non-polar	0

Table 3.2: Numerical representation for possible amino acid side-chain polarity

3.2.3 Charge

In regards to side-chain charge, amino acids are divided into three groups; positive, negative and neutral as shown in Table 2.1. These properties along with hydrophilicity or hydrophobicity are important for protein structure and protein–protein interactions [11]. Therefore we have decided to use amino acids' charge as a feature. The Histidine H amino acid is positive (10%) neutral (90%) and thus we considered it to be neutral. A window of size w would give a w number of features representing each amino acid included by the window. Same as we did with polarity, we have added an extra feature called window charge to represent the overall charge of the window. Window charge c is determined by summing the charge of each residue included by the window then if the sum is greater than or equal to zero, the window charge is set to '1' and '-1' otherwise as shown in 4. The numerical representations for all possible amino acid side-chain charge are shown in Table 3.3.

$$c = \begin{cases} 1, & \text{if sum of residues charge} \geq 0 \\ -1, & \text{otherwise} \end{cases} \tag{4}$$

Side-chain charge	Numerical representation
Positive	1
Negative	-1
Neutral	0

Table 3.3: Numerical representation for possible amino acid side-chain charge

3.2.4 Hydropathy and Hydrophilicity

Amino acids have different hydropathy and hydrophilicity indices influencing their location inside the protein. The water-soluble proteins tend to have their hydrophobic residues (Leu, Ile, Val, Phe, and Trp) buried in the middle of the protein, whereas hydrophilic side-chains are exposed to the aqueous solvent [40]. Two sets of features each equals the given window size were extracted by replacing each amino acid with its corresponding hydropathy and hydrophilicity indices. Moreover, we have added two extra features representing the averages of window hydropathy and window hydrophilicity. Hydropathy and hydrophilicity indices for the 20 amino acids are listed in Table 2.1.

3.2.5 Corresponding Secondary Structure Sequence

Protein secondary structure can provide some useful features. To determine the protein's secondary structure we first sent the protein sequence to the

PSIPRED Protein Sequence Analysis Workbench operated by UCL Department Of Computer Science. The predicted secondary structure sequence consists of three types of structures. Each amino acid in the protein sequence is represented by either an C, E or H letter on the secondary structure sequence where C stands for coil while H stands for α-helix and E stands for β-sheet as shown in Figure 3.5. The number of features extracted in this step is equal to the window size. The numerical representations for all possible secondary structure shapes are shown in Table 3.4. In addition, we have added one more feature representing Window secondary structure s. This feature was determined by summing the secondary structure of each residue included by the window then if the sum is greater than or equal to zero, the window charge is set to '1' and '-1' otherwise as shown in 5.

$$s = \begin{cases} 1, & \text{if sum of residues secondary structure} \geq 0 \\ -1, & \text{otherwise} \end{cases} \tag{5}$$

```
   210                                   250
---PYARPNAIVGTDASRNVITLGGTRAELENYLRTVQIFDV---
---CCCCCCCEEEEECCCCEEEECCCHHHHHHHHHHHHHCCC---
```

Figure 3.5: Sample residues of the XpsD protein and their corresponding secondary structure

3.2.6 Amino Acids Occurrences

In this step we recorded the occurrences of each amino acid included by the window producing a new set of 20 features. Each feature represents the total number of occurrences for one amino acid.

Secondary structure	Numerical representation
C	0
H	1
E	-1

Table 3.4: Numerical representation for possible secondary structure shapes

3.2.7 Presence of Glycine

The Glycine is known to be the most flexible amnio acid, therefore its presence may increase any chance of successful insertion. To benefit from this phenomena we have been able to extract two more features in addition to total number of Glycine occurrences inside the window. The total number of Glycine occurrences was determined by the previous step. The two features indicate if the amino acid directly before or after the insertion site belong to Glycine. These two features can take the value of '1' or '0'.

3.2.8 Feature Vector

During this step we constructed our final feature vector where, each insertion site is represented by a feature vector created by combining all features extracted in the previous steps. The length of any feature vector depends solely on the chosen window size. Refere to Table 3.5 for a list of all extracted features and their possible numerical representations.

A window of size 7 will generate a total of 143 features arranged as shown in Table 3.6.

Feature set	Features #	Possible values
Dipeptide composition	81	[0, 1]
Polarity	w	{0, 1}
Window polarity	1	{0, 1}
Charge	w	{-1, 0, 1}
Window charge	1	{-1, 1}
Hydropathy index	w	column 5 in Table 2.1
Window hydropathy avg.	1	[-4.5, 4.5]
Hydrophilicity index	w	column 6 in Table 2.1
Window hydrophilicity avg.	1	[-3.98, 15.86]
Secondary structure	w	{-1, 0, 1}
Window secondary structure	1	{-1, 1}
Amino acids occurrences	20	[0, w]
Neighboring Glycine	2	{0, 1}
Class label	1	{P, N}[1]

[1] P = Permissive, N = Non-permissive

Table 3.5: List of all extracted features and their possible numerical representations

3.3 Dataset Collection

The presence of a well known and documented dataset is essential for every classification problem. A great deal of effort in this work was directed towards constructing a training dataset for protein insertion sites as there is no dataset for protein insertion sites currently available. We have constructed our own dataset by collecting individual protein sequences from a number of different biology publications. We had to scan each publication for insertion sites and identify both the permissive and non-permissive insertion sites.

3.4 Classifier Training

Continuing with our methodology, we have trained an SVM classifier with 5-fold cross-validation to be later used for predicting permissive insertion

Feature set	Features #
Dipeptide composition	81
Polarity	7
Window polarity	1
Charge	7
Window charge	1
Hydropathy index	7
Window hydropathy avg.	1
Hydrophilicity index	7
Window hydrophilicity avg.	1
Secondary structure	7
Window secondary structure	1
Amino acids occurrences	20
Glycine before	1
Glycine after	1

Table 3.6: Possible number of features generated for $w = 7$

sites. When a classifier is trained a model is created, this model can then be applied for any dataset to predict permissive insertion sites.

Chapter 4

4 EXPERIMENTS AND RESULTS

4.1 Dataset

In our approach, we have tested our system with real world data as mentioned in the previous chapter. Due to the lack of sufficient number of semi-permissive sites samples; all semi-permissive sites were considered as permissive in the training set.

For the train and test method, the available data is split into two parts called a training set and a test set (Figure 4.1). First the training set is used to construct the SVM classifier. The classifier is then used to predict the classification for the instances in the test set. If the test set contains N instances of which C are correctly classified the prediction accuracy of the classifier for the test set is $p = C/N$. This can be used as an estimate of its performance on any unseen dataset.

4.2 Training and Validation

Training involves using a dataset with known values, and learning a model from that dataset. However, models that fit the training dataset very well may fail to predict new data points. Such over-fitting of the training data will most likely yield a model that cannot be generalized and, therefore, would

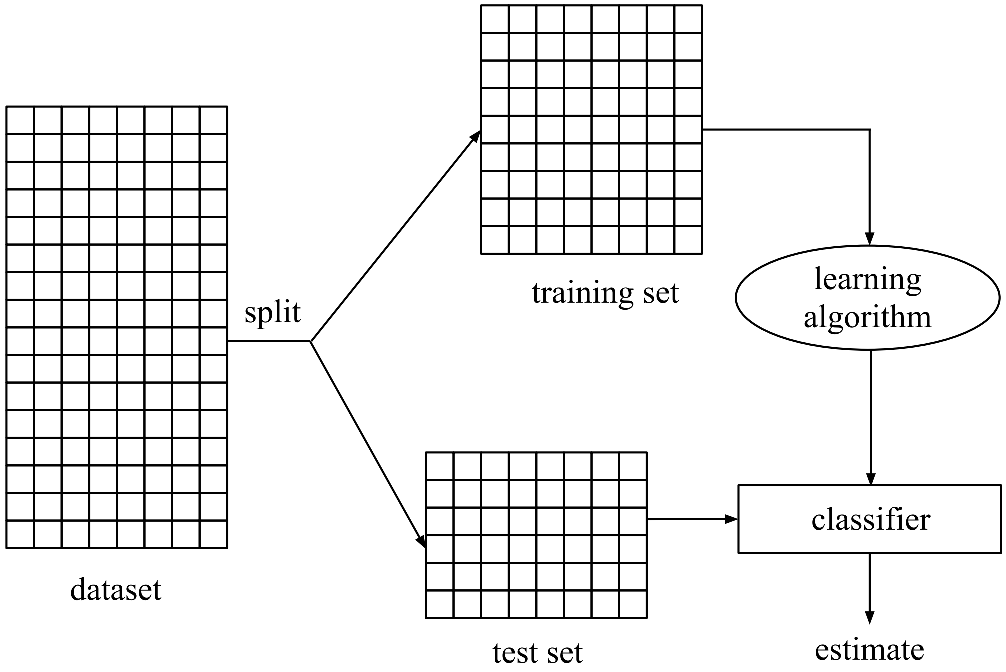


Figure 4.1: Train and test [7]

not be useful. Therefore, an algorithm and its associated parameters must be validated before they are used to predict new data. This process involves segmenting the training data into two sets. One set is used for training and the other for testing the model. Typically, validation should be done with a variety of algorithms and parameters, and results monitored to choose the best combination. This combination can then be used to build a model with the entire training dataset, and subsequently to predict for new data. Cross-validation is an important tool to avoid overfitting models on training data, as overfitting will give low accuracy on validation. Also, validation can help choose the right set of descriptors, an appropriate algorithm and associated parameters for a given dataset. Validation can be run on the same dataset using various algorithms and altering the parameters of each

References	UniProt code[1]	Insertions #	P[2]	N[3]
Manoil et. al. [24]	LACY_ECOLI	21	10	11
Bailey et. al. [3]	BGAL_ECOLI	8	6	2
Schlehuber and Rose [34]	VGLG_VSIVA	6	6	0
Charbit et. al. [9]	LAMB_ECOLI	13	7	6
Guedin et. al. [14]	FHAC_BORPE	18	10	8
Nelson and Traxler [28]	MALG_ECOLI	12	2	10
Teymournejad et. al. [37]	FMS3_ECOLX	3	3	0
Lippincott and Taxler [23]	MALK_ECOLI	12	7	5
Haft et. al. [15]	TRAI1_ECOLI	33	21	12
Lee et. al. [22]	TRAD1_ECOLI	9	3	6
Total # of samples		135	75	60

[1] Protein Code on The Universal Protein Resource (UniProt)
[2] Permissive sites #
[3] Non-permissive site #

Table 4.1: Training dataset

References	UniProt code	Insertions #	P	N
Schifferli and Alrutz [33]	FASD_ECOLX	17	7	10
Nelson et. al. [27]	LACI_ECOLI	18	8	10
N.T. Hu and Others [17]	GSPD_XANCP	21	9	12
Lecroisey et. al. [21]	A7ZUQ6_ECO24	10	8	2
Total # of samples		66	32	34

Table 4.2: Test dataset

algorithm. The results of validation can then be examined to choose the best algorithm and parameters for the model.

During N-Fold Cross-Validation, the compounds in the input data are randomly divided into N equal parts; N-1 parts are used for training, and the remaining 1 part is used for testing. The process is repeated N times, with a different part being used for testing in each iteration. Thus, each compound is used at least once in training and once in testing, and the average results are reported. This whole process is then repeated as many times as specified by the ‘number of repeats’ [25] Figure 4.2.

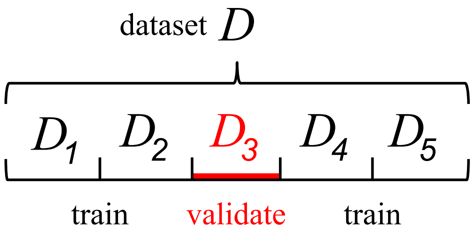


Figure 4.2: 5-fold cross-validation

4.3 Experiments with 5-Fold Cross-Validation

In our work, we used LIBSVM classifier available within the WEKA toolkit. From the available user specific parameters associated with the LIBSVM classifier, we changed the cost parameter C. The default value for this parameter was 1 and we changed it to 30. SVM models have a cost parameter which, allows some flexibility in separating the categories by controlling the trade off between allowing training errors and forcing rigid margins. It creates a soft margin that permits some misclassifications. Increasing the value of C increases the cost of misclassifying points and forces the creation of a more accurate model that may not generalize well. We also enabled the normalization option. Large margin classifiers are known to be sensitive to the way features are scaled. Therefore it is essential to normalize either the data or the kernel itself. This observation carries over to kernel-based classifiers that use non-linear kernel functions: The accuracy of an SVM can severely degrade if the data is not normalized. Normalization can be performed at the level of the input features or at the level of the kernel (normalization in feature space).

We ran the algorithm with five-fold cross-validation and for different window sizes. Our training set consisted of 135 samples 75 of which are labeled as permissive and the remaining 60 labeled as non-permissive.

w	Features #	Accuracy (%)	RMSE
7	143	48.15	0.7201
9	153	52.59	0.6885
11	163	52.59	0.6885
13	173	56.30	0.6611
15	183	57.78	0.6498

Table 4.3: Training set results for different window sizes using SVM classifier with 5-fold cross-validation

The Root Mean Square Error RMSE is a measure of the differences between values predicted by a model or an estimator and the values actually observed from the thing being modeled or estimated. RMSE is also a a measure of how well the curve fits the data. Since the RMSE is a good measure of accuracy, it is ideal if it is small.

The results we obtained from each window size are shown in Table 4.3 and the appropriate confusion matrices for all proteins and window sizes are displayed in Table 4.5.

4.4 Experiments with Feature Selection

We performed feature selection using the two prominent methods; the wrapper and filter. After performing the feature selection, we trained two SVM classifier models on the same training set described in Table 4.1. One model was trained using the set of features selected by the wrapper method and

	Accuracy (%)			
w	FasD[1]	LacI[2]	XpsD[3]	MalE[4]
7	64.71	61.11	42.86	80.00
9	58.82	44.44	47.62	70.00
11	41.18	44.44	42.86	70.00
13	64.71	55.56	47.62	50.0
15	58.82	50	52.31	60.00

[1] Outer membrane usher protein FasD
[2] Lactose operon repressor LacI
[3] Type II secretion system protein XpsD
[4] Maltose periplasmic protein MalE

Table 4.4: Test set results for different window sizes using SVM trained model

	w										
	7		9		11		13		15		
Protein	P	N	P	N	P	N	P	N	P	N	
FasD	6	1	4	3	3	4	6	1	6	1	P
	5	5	4	6	6	4	5	5	6	4	N
LacI	4	7	5	3	4	4	5	3	5	3	P
	3	4	7	3	6	4	5	5	6	4	N
XpsD	4	5	3	6	3	6	6	3	6	3	P
	7	5	5	7	6	6	8	4	7	5	N
MalE	6	2	5	3	5	3	5	3	6	2	P
	0	2	0	2	0	2	2	0	2	0	N

Table 4.5: Confusion matrices of test set results for different window sizes using SVM trained model

another using the features selected by the filter method. The number of selected features and the accuracy of the SVM classifiers for both feature selection methods and for different window sizes are shown in Table 4.6 and 4.8.

4.4.1 Wrapper Method

We used the WEKA toolkit to compute the feature selection subsets using an attribute selector called ClassifierSubsetEval. The wrapper method uses a subset evaluator which creates all possible subsets from the feature vector. Then it uses a classification algorithm to induce classifiers from the features in each subset. Finally, it will consider the subset of features with which the classification algorithm performs the best.

4.4.2 Filter Method

For the filter method we have also used the WEKA toolkit to compute the feature selection subsets by choosing an attribute selector called InfoGainAttributeEval. Unlike the wrapper method the fileter method does not require a classifier to select useful features it rather uses a ranker algorithm. The ranker algorithms are used to rank the features by omitting one feature at a time from the rank list. From the list of ranked features we choose the ones that have rank value > 0.

4.5 Results

Our experiments produced two sets of results, a set without feature selection and another with feature selection. Starting with the results obtained before the feature selection, we have the training set results and the test set results. As for the training set results Table 4.3, first we notice the proportional relationship between the number of features and w. Regarding the accuracy of the SVM classifier all window sizes produced fairly good accuracies apart from $w = 7$ which, produced a lower accuracy. Highest accuracy was achieved

w	#[1]	Index[2]	Description[3]	Accuracy (%)	RMSE
7	4	11	DC(11)	57.78	0.6498
		94	Charge(5)		
		100	Hydropathy(3)		
		107	Hydrophilicity(2)		
9	10	14, 63, 67, 68	DC(14, 63, 67, 68)	66.67	0.5774
		86	Polarity(5)		
		96, 97, 100	Charge(5, 6, 9)		
		110	Hydropathy(9)		
		114	Hydrophilicity(3)		
11	6	8, 14, 39, 44	DC(8, 14, 39, 44)	59.26	0.6383
		112	Hydropathy(7)		
		121	Hydrophilicity(4)		
13	4	56	DC(1)	71.11	0.5375
		136	Hydrophilicity(13)		
		141	Secondary Structure(4)		
		152	Alanine count		
15	6	10, 75	DC(10, 75)	67.41	0.5709
		116	Hydropathy(3)		
		143	Hydrophilicity(14)		
		147	Secondary Structure(2)		
		178	Theonine count		

[1] Number of selected features
[2] Feature index in the feature vector
[3] Feature name

Table 4.6: Training set results for different window sizes using SVM classifier with 5-fold cross-validation after applying wrapper method feature selection

with $w = 15$. However experiments proved that a higher accuracy does not necessarily mean a better model. In other words, achieving high accuracy on the training set could be a sign of overfitting. Overfitted models do not generalize well and hence perform poorly when used for predicting unseen data.

	Accuracy (%)			
w	FasD	LacI	XpsD	MalE
7	52.94	61.11	61.90	70.00
9	52.94	50	52.38	50.00
11	52.94	55.56	47.62	80.00
13	52.94	61.11	52.38	60.00
15	47.06	61.11	38.06	50.00

Table 4.7: Test set results for different window sizes using SVM trained model after applying wrapper method feature selection

The trained SVM model was applied to the test data set described in Table 4.2. The test set consisted of four different proteins. Proteins were individually classified using the trained model. The process was repeated for different window sizes to be able to compare and choose the optimal w.

The results listed in Table 4.4 show that although $w = 7$ produced the lowest accuracy on the training set, it outperformed all other window sizes and demonstrated a better performance on the test set. However, $w = 15$ yielded the highest classification accuracy for the type II secretion system protein XpsD. Moreover, $w = 11$ provided the lowest classification accuracy for the outer membrane usher protein FasD . The reason for this can be related to the increased number of features and hence the increased possibility of noise.

Results for training with different window sizes after applying the feature selection or wrapper subsets are given in Table 4.6. It can be observed that, the number of features has been significantly decreased. Also, the accuracy of the trained SVM model has significantly increased for all window sizes. This is expected because feature selection efficiently excludes most noisy features. The most increased accuracy was recorded for $w = 13$, where it

w	#	Index	Description	Accuracy (%)	RMSE
7	3	19	DC(19)	63.70	0.6025
		83	Polarity(2)		
		107	Hydrophilicity(2)		
9	3	19	DC(19)	66.67	0.5774
		84	Polarity(3)		
		114	Hydrophilicity(3)		
11	4	12	DC(12)	61.48	0.6206
		85	Polarity(4)		
		121	Hydrophlicity(4)		
		144	Aspartic acid count		
13	11	14, 19	DC(14, 19)	68.15	0.5644
		86	Polarity(5)		
		122	Hydropathy(13)		
		123	Hydropathy Avg.		
		128, 136	Hydrophilicity(5, 13)		
		137	Hydrophilicity Avg.		
		154	Aspartic acid count		
		162	Methionine count		
		168	Threonine count		
15	10	12, 14	DC(12, 14)	65.19	0.59
		87	Polarity(6)		
		127	Hydropathy(14)		
		135, 143	Hydrophilicity(6, 14)		
		145	Hydrophilicity Avg.		
		164	Aspartic acid count		
		172	Methionine count		
		178	Threonine count		

Table 4.8: Training set results for different window sizes using SVM classifier with 5-fold cross-validation after applying filter method feature selection

has increased by almost 15%. The selected features column holds significant information about what features have actually been selected. Moreover, the

	Accuracy (%)			
w	FasD	LacI	XpsD	MalE
7	70.59	50	47.62	60.00
9	58.82	50	52.38	60.00
11	41.18	44.44	52.38	70.00
13	47.06	44.44	33.33	80.00
15	52.94	27.78	28.57	90.00

Table 4.9: Test set results for different window sizes using SVM trained model after applying filter method feature selection

nature of the selected features is a good indicator of how relevant our choice of features is. It can be observed, that dipeptide composition features are common among all window sizes. This is a very important indicator that our choice of features is smart. Regarding the other feature sets, it was noticed that all window sizes have at least one hydropathy or hydrophilicity feature. Unexpectedly, features such as the ones related to Glycine were never selected. This could indicate that hudrophilicity and hydropathy related features hold more significant information than the Glycine related ones. Moreover, Secondary structure related features were only selected once.

The classification results after the wrapper method feature selection which are listed in Table 4.7, show that $w = 7$ produced the best overall accuracy. Moreover, the XpsD protein has witnessed the highest prediction accuracy. Accuracy of predicting permissive sites for the FasD has not improved after the feature selection, instead it has slightly dropped.

Training results after the filter method feature selection are shown in Table 4.8. Training accuracy for all window sizes was relatively close, however, $w = 13$ recorded the highest training accuracy. Observing the selected features column at the same table, it can be noticed that the dipeptide composition along with the hydrophiliciy related features are the most prominent features as they were selected for all different window sizes. Our choice of hydropathy and hydrophilicity average features was smart, were these features got selected in two cases. Surprisingly, none of the secondary structure related features were selected. It is also worth mentioning that in the case of the amino acid occurrences features, the Asparatic acid D and Threonine T, both hydrophilic were selected in three cases. This supports the fact that hydrophilic regions of the protein have a higher potential of including permissive sites. Methionine M was also selected on two cases.

Results for testing the prediction model after applying the filter method feature selection are listed in Table 4.9. Again $w = 7$ gave the highest overall classification results. The FasD and MalE proteins have seen the best classification results with $w = 7$ and $w = 15$ respectively. However the rest of the proteins did not show any improvement instead the classification accuracy significantly dropped as in the case of the LacI and XpsD proteins at $w = 15$. This is the result of overfitted model. The model started to show overfitting signs as the w increased.

It is extremely important to note that our training and test sets are comprised of a variety of proteins belonging to different protein families. Therefore, classification results varies each time we add or remove proteins from the training dataset. There are proteins that perform well together and produce a very accurate prediction model, but still the performance of the model depends on the proteins in the test dataset and how close they are to the proteins in the training set. For example, training using LacY or LacI to test on FasD did not yield good results as FasD belongs to different protein family with different characteristics. However, training XpsD to predict

FasD yielded better results. We strongly believe that this issue is worth more investigation. If this is proven to be the case, then we suggest building multiple trained models each trained using proteins belonging to different protein family and in this situation a suitable model is chosen depending on the family of the protein to be predicted for permissive sites.

Precision and recall are measures of performance and can help when trying to classify very skewed classes, where one class is rare in the data. Simply taking a percentage of correct classifications can be misleading, since always guessing the more common class means the classifier will almost always be right. However, the ratio of permissive to non-permissive classes in both the training and test datasets is almost 1:1. Therefore we did not record the values for precision and recall.

The training dataset being relatively small is that main reason that our trained model showed some signs of overfitting.

Chapter 5

5 CONCLUSION AND FUTURE WORKS

In this work, we have introduced an innovative way for effectively extracting features to predict permissive insertion sites. We have investigated the effect of feature selection and the resulting classification performance. More specifically, feature subsets determined with a wrapper method. Extensive experiments performed with different window sizes, lead to the following conclusions.

Our educated guess approach was indeed effective. This claim was strongly supported by the nature of the features that was selected after applying both the wrapper and filter methods. Features from all feature sets except the Glycine related ones, were proven to be useful. Unexpectedly, protein secondary structure related features were proven to be redundant as only one protein secondary structure feature appeared once with $w = 13$.

The classification accuracy achieved with the feature selection was higher than the accuracy achieved with the complete feature set and with a reduction of at least 95% of the feature space for all window sizes.

Despite the limited number of training samples we have been able to predict permissive sites with very high accuracy as in the case of MalE protein.

Our experiment shows that the prediction of permissive sites is protein family sensitive. This means that it could be better to have in the training dataset, proteins that belong to the same protein family as the proteins in the test dataset. This issue deserves further investigation and study.

Our work is at a very early stage, so there is a place for future work and improvement. Perhaps, one of the main obstacles that limited the performance of our system was the relatively small size of the training dataset. Therefore, increasing the size of the training dataset should be a priority for any future work.

The results of our experiment show that some of the extracted features did not improve the classification accuracy and even proven to be totally redundant at some point. This fact, makes it imperative to start thinking of alternative sets of features. One new set of features could be constructed by using a dictionary of words collected from previously identified permissive sites. Features could be extracted by segmenting the amino acids in the window to possible words and then compare the words obtained from the window against the words in the dictionary and record the fractions as features.

References

[1] Carolyn F. Allex, Jude W. Shavlik, and Frederick R. Blattner. Neural network input representations that produce accurate consensus sequences from dna fragment assemblies. *Bioinformatics*, 15(9):723--728, 1999.

[2] R. Aroul-Selvam, Tim Hubbar, and Rajkumar Sasidharan. Domain insertions in protein structures. *J Mol Biol*, 338(4):633--641, 2004. doi:10.1016/j.jmp.2004.03.039.

[3] Jeannie Bailey and Colin Manoil. Identification of permissive and non-permissive peptide insertion sites in β-galactosidase using the ez::tntm in-frame linker insertion kit. *Journal Of Bacteriology*, 8(1):1--3, 2001.

[4] Timothy L. Bailey and William Noble Grundy. Classifying proteins by family using the product of correlatedp-values. In *Proceedings of the Third Annual International Conference on Computational Molecular Biology.*

[5] Avrim L. Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245--271, 1997.

[6] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *COLT '92 Proceedings of the fifth annual workshop on Computational learning theory*, pages 144--152. Pittsburgh, PA, 1992. ACM Press, 1992.

[7] Max Bramer. *Principles of Data Mining.* Springer, 1st edition, 2007.

[8] Carl-Ivar Branden and John Tooze. *Introduction to Protein Structure.* Garland Publishing, 2nd edition, 2010.

[9] Alain Charbit, Jorge Ronco, Valerie Michel, Catherine Werts, and Maurice Hofnung. Permissive sites and topology of an outer membrane protein with a reporter epitope. *J. Bacteriol.*, 173(1):262--275, Jan 1991.

[10] Nello Christianini and John Shawe-Taylor. *An introduction to support vector machines: and other kernal-based learning methods.* Cambridge University Press, Cambridge U.k. ; New York, 2000.

[11] Thomas E. Creighton. Chapter 1. In *Proteins: Structures and Molecular Properties.* W. H. Freeman, San Francisco, 2nd edition, 1993.

[12] Nello Cristianini and John Shawe-Taylor . *An introduction to support Vector Machines: and other kernel-based learning methods.* Cambridge University Press, New York, NY, USA, 2000.

[13] Sanmay Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *Proceedings of the Eigh- teenth International Conference on Machine Learning*, pages 74--81, 2001.

[14] Sandrine Guedin, Eve Willery, Jan Tommassen, Emmanuelle Fort, Herve Drobecq, Camille Locht, and Francoise Jacob-Dubuisson. Novel topological features of FhaC, the outer membrane transporter involved in the secretion of the Bordetella pertussis filamentous hemagglutinin. *J. Biol. Chem.*, 275(39):30202--30210, Sep 2000.

[15] Rembrandt J. F. Haft, Gilberto Palacios, Tran Nguyen, Manuela Mally, Eliora G. Gachelet, Ellen L. Zechner, and Beth Traxler. General mutagenesis of F plasmid TraI reveals its role in conjugative regulation. *J. Bacteriol.*, 188(17):6346--6353, Sep 2006.

[16] Karin L Heckman and Larry R Pease. Gene splicing and mutagenesis by pcr-driven overlap extension. *Nature Publishing Group*, 2(4), 2007. doi:10.1038/nprot.2007.132.

[17] Nien-Tai Hu, Ming-Ni Hung, David Chanhan Chen, and Rong-Tzong Tsai. Insertion mutagenesis of xpsd, an outer- membrane protein involved in extracellular protein secretion in xanthomonas campestris pv. campestris. *Microbiology*, 144:1479--1486, 1998.

[18] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273--324, 1997.

[19] Jack Kyte and Russell F. Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 157(1):105--132, 1982.

[20] Pat Langley. Selection of relevant features in machine learning. In *In Proceedings of the AAAI Fall symposium on relevance*, pages 140--144. AAAI Press, 1994.

[21] Anne Lecroisey, Pierre Martineau, Maurice Hofnung, and Muriel Delepierre. NMR studies on the flexibility of the poliovirus C3 linear epitope inserted into different sites of the maltose-binding protein. *J. Biol. Chem.*, 272(1):362--368, Jan 1997.

[22] Martin H. Lee, Nick Kosuk, Jeannie Bailey, Beth Traxler, and Colin Manoil. Analysis of F factor TraD membrane topology by use of gene fusions and trypsin-sensitive insertions. *J. Bacteriol.*, 181(19):6108--6113, Oct 1999.

[23] John Lippincott and Beth Traxler. MalFGK complex assembly and transport and regulatory characteristics of MalK insertion mutants. *J. Bacteriol.*, 179(4):1337--1343, Feb 1997.

[24] Colin Manoil and Jeannie Bailey. A simple screen for permissive sites in proteins: Analysis of escherichia coli lac permease. *J. Mol. Biol.*, 267:250--263, 1997.

[25] Tom M. Mitchell. §4.6.5 in machine learning. In *Generalization, Overfitting, and Stopping Criterion*, pages 111--112. W. H. Freeman, mcgraw-hill edition, 1997.

[26] Nomenclature and symbolism for amino acids and peptides. *Pure Appl. Chem.*, 56(5):595--624, 1984. doi:10.1351/pac198456050595.

[27] Bryn D. Nelson, Colin Manoil, and Beth Traxler. Insertion mutagenesis of the lac repressor and its implications for structure-function analysis. *Journal of Bacteriology*, 179(11):3721--3728, 1997.

[28] Bryn D. Nelson and Beth Traxler. Exploring the role of integral membrane proteins in ATP-binding cassette transporters: analysis of a collection of MalG insertion mutants. *J Bacteriol*, 180(9), 1998.

[29] Eugene W. Nester, Denise G. Anderson, Jr. C. Evans Roberts, and Martha T. Nester. *Microbiology A Human Perspective*. McGraw-Hill, 5th edition, 2007.

[30] Yanay Ofran and Burkhard Rost. Predicted protein-protein interaction sites from local sequence information, 2003.

[31] Pasquale Petrilli. Classification of protein sequences by their dipeptide composition. *Bioinformatics*, 9(2):205--209, 1993.

[32] Ahmet Saçan, Özgür Öztürk, Hakan Ferhatosmanoğlu, and Yusu Wang. Lfm-pro: a tool for detecting significant local structural sites in proteins. *Bioinformatics*, 23(6):709--716, 2007. ISSN 1367--4803.

[33] Dieter M. Schifferli and Michael A. Alrutz. Permissive linker insertion sites in the outer membrane protein of 987p fimbriae of escherichia coli. *Journal of Bacteriology*, 176(4):1099--1110, 1993.

[34] Lisa D. Schlehuber and John K. Rose. Prediction and identification of a permissive epitope insertion site in the vesicular stomatitis virus glycoprotein. *Journal of Virology*, 78(10):5079--5087, 2004.

[35] Bernard Schölkopf, Koji Tsuda, and Jean-Philippe Vert. *Kernel Methods in Computational Biology*. MIT Press, 2004.

[36] Carlos Setubal and Joao Meidanis. *Introduction to Computational Molecular Biology*. PWS, 1997.

[37] Omid Teymournejad, Ashraf M. M]obarez, Zuhair M. Hassan, Seyed M. Moazzeni, Bagher Yakhchali, and Vajihe Eskandari. In silico prediction of exposure amino acid sequences of outer inflammatory protein A of Helicobacter pylori for surface display on Eschierchia coli. *Indian J Hum Genet*, 18(1):83--86, Jan 2012.

[38] Gerard J. Tortora, Berdell R. Funke, and Christine L. Case. *Microbiology an Introduction*. Benjamin Cummings, USA, 10th edition, 2010.

[39] Alican Türk and O. Uğur Sezerman. Predict permissive sites of protein for insertion domain, 2010.

[40] Dan W. Urry. The change in gibbs free energy for hydrophobic association: Derivation and evaluation by means of inverse temperature transitions. *Chemical Physics Letters*, 399(1-3):177--83, 2004. doi:10.1016/S0009-2614(04)01565-9.

[41] Jason T. L. Wang, Qicheng Ma, Dennis Shasha, and Cathy H. Wu. New techniques for extracting features from protein sequences. *IBM Systems Journal*, 40(2):426--441, 2001.

[42] Leslie Wilson, Paul Matsudaira, and Alan Tartakoff. *Vectorial Transport of Proteins into and across Membranes*. Academic Press, 1st edition, 1991.