



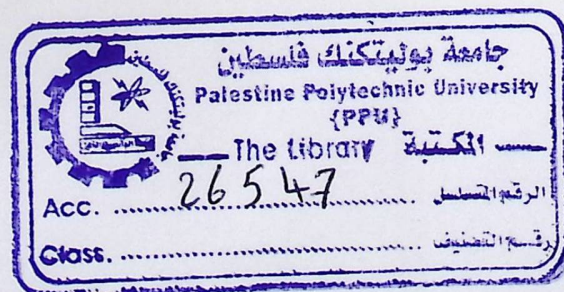
Palestine Polytechnic University
Deanship of Graduate Studies and Scientific Research
Master of informatics

Using Clustering to Enhance Protein Sequence Classification

Submitted by:

Haneen Musallam Altartouri

Thesis submitted in partial fulfillment of requirements of the
degree Master of Science in Informatics
May, 2013



DECLARATION

I declare that the Master Thesis entitled "Using Clustering to Enhance Protein Sequence Classification" is my original work, and hereby certify that unless stated, all work contained within this thesis is my own independent research and has not been submitted for the award of any other degree at any institution, except where due acknowledgement is made in the text.

Haneen Musallam Hussein Altartouri

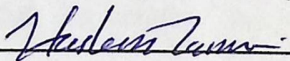
Signature: _____ *Sup*

Date: *14 / July / 2013*

The undersigned hereby certify that they have read, examined and recommended to the Deanship of Graduate Studies and Scientific Research at Palestine Polytechnic University the approval of a thesis entitled: **Using Clustering to Enhance Protein Sequence Classification**, submitted by **Haneen M. Altartouri** in partial fulfillment of the requirements for the degree of Master in Informatics.

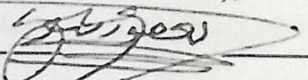
Graduate Advisory Committee:

Dr. Hashem Tamimi (Supervisor), Palestine Polytechnic University.

Signature: 

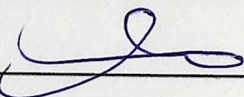
Date: 20/7/2013

Dr. Yaqoub Ashhab (Supervisor), Palestine Polytechnic University.

Signature: 

Date: 5 July -2013

Dr. Mohammed Aldasht (Internal committee member), Palestine Polytechnic University.

Signature: 

Date: 8 July / 2013

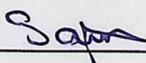
Dr. Mohamed Alshalalfa (External committee member), University of Calgary.

Signature: 

Date: 3/July/2013

Thesis Approved

Dr. Sameer Khader Dean of Graduate Studies and Scientific Research Palestine Polytechnic University

Signature: 

Date: 21.07.2013

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for the master degree in Informatics at Palestine Polytechnic University, I agree that the library shall make it available to borrowers under rules of the library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of the source is made.

Permission for extensive quotation from, reproduction, or publication of this thesis may be granted by my main supervisor, or in his absence, by the Dean of Graduate Studies and Scientific Research when, in the opinion of either, the proposed use of the material is for scholarly purposes.

Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Haneen Musallam Hussein Altartouri

Signature: _____

Date: 14 July 2013

ACKNOWLEDGEMENT

DEDICATION

As I write the last words of this thesis, I greatly appreciate the thesis's
*To my mother and father, who always supported me and encouraged me to
get my Master degree. To my brothers and sisters for their support.*

I would like to thank the thesis's examiner Dr. Mohamed Alsharafa and
Dr. [Name] Alkhatib for their valuable suggestions and corrections to
the work, which greatly helped me to improve in various aspects.

I would like to thank my family for continued support, encouragement
and patience from the first step till the end.

ACKNOWLEDGEMENT

As I write the last words of this thesis, I greatly appreciate the thesis's supervisors Dr. Hashem Tamimi and Dr. Yaqoub Ashhab for their support and time they spent with me in order for this thesis to succeed.

I would like to thank the thesis's examiner Dr. Mohamed Alshalalfa and Dr. Mohammed Aldasht for their valuable suggestions and corrections to this work, which greatly helped me to improve in various aspects.

I would like to thank my family for continued support, encouragement and patience from the first step till the end.

الملخص

تهدف هذه الرسالة الى تحسين اداء عملية التنبؤ بالسماط البيولوجية لسلاسل البروتينات بالاعتماد على خوارزميات التصنيف و التجميع معا، حيث يتم تجميع سلاسل البروتينات الى مجموعات بناء على درجة التشابه بينهم و ذلك باستخدام احد خوارزميات التجميع و من ثم يتم تطبيق خوارزميات التصنيف على كل مجموعة. النموذج المقترح مناسب ايضا لقواعد البيانات الكبيرة التي تتطلب دقة و سرعة في التنبؤ.

في هذا النموذج تم استخدام عدة خصائص فيزيائية و كيميائية للحموض الامينية بعض هذه الخصائص هي خصائص أصلية و بعضها الاخر هي خصائص مشتقة من الخصائص الاصلية. بالإضافة الى ذلك، تم استخدام طريقتين لتمثيل الحموض الامينية على شكل ارقم بالاعتماد على الخصائص السابقة. هذه الخصائص و طرق التمثيل تم اختيارهم لتحسين اداء النموذج المقترح.

لاختبار النموذج المقترح تم فحصه هلى ثلاثة انواع مختلفة من سلاسل البروتينات، ومن خلال النتائج التي تم التوصل اليها نلاحظ ان تجميع سلاسل البروتينات قبل تصنيفها يعطي نتائج افضل مقارنة باستخدام التصنيف لوحده من حيث الدقة في التنبؤ . بالإضافة الى ذلك اظهرت النتائج المتعلقة بأداء الوقت ان النموذج المقترح نجح في تقليل الوقت اللازم لترتيب خوارزمية التصنيف بشكل ملحوظ ، و هذا يدل على قدرة النظام المقترح على التعامل مع قواعد البيانات الكبيرة دون الحاجة لحذف أي جزء منها.

Abstract

We introduce a new approach for enhancing the performance of prediction of biological attributes based on protein sequences using a combination of classification algorithms and clustering analysis. Before applying classification, we use clustering analysis in order to find clusters of similar proteins. A classification algorithm is then applied on each cluster. The proposed approach is suitable for large datasets, when high classification accuracy and fast convergence are required.

Different descriptors based on the physicochemical properties of amino acids are used, some of them are native properties and the others are derived properties. Two encoding methods are used to represent the protein sequences using the descriptors. These descriptors and encoding methods are analyzed to enhance the performance of the proposed approach.

Three standard benchmark datasets, Caspase, Major Histocompatibility Complex class II (MHC-II) and the membrane proteins are used to examine the proposed approach. Many experiments with different parameters are performed and the results are cross validated.

The results show that applying clustering prior to classification gives higher prediction accuracy than using the classification without clustering, especially when using the membrane proteins dataset and the Caspase dataset. In addition, the result of time performance, especially when using the MHC-II

dataset, shows that the proposed approach succeeds in reducing the training time of the classification algorithm significantly while maintaining the accuracy of prediction. That means our approach can handle large datasets, without the need to reduce the data.

TABLE OF CONTENTS

Table of Contents

1.1.1	Selection of active amino acids PCPs	31
1.1.2	Definition of novel descriptors	32
1.2	Classification based on clustering	35
1.4	Thesis contribution	36
1.4.1	Membrane proteins benchmark	39
1.4.2	C class II benchmark	39
1	Introduction	2
1.1	Thesis objective	4
1.2	Contributions	4
1.3	Thesis Organization	5
2	Background	6
2.1	Physicochemical properties of amino acids	6
2.2	Encoding the protein sequences	7
2.2.1	Encoding methods using the amino acid sequence	7
2.2.2	Encodings using PCPs of amino acids	10
2.3	Machine Learning Techniques	18
2.3.1	Cluster Analysis	18
2.3.2	Classification	20
2.4	Feature selection and reduction techniques	27
2.4.1	Feature selection	27
2.4.2	Feature extraction	28
3	Literature Review	30
3.1	Importance of encoding protein sequences using the physico-chemical properties	30
3.2	Representing the amino acids based on PCPs	31

TABLE OF CONTENTS

Table of Contents

1.1.1	Selection of active amino acids PCPs	31
1.1.2	Definition of novel descriptors	32
1.2	Classification based on clustering	35
1.3	Thesis contribution	36
1.3.1	Membrane protein benchmark	39
1	Introduction	2
1.1	Thesis objective	4
1.2	Contributions	4
1.3	Thesis Organization	5
2	Background	6
2.1	Physicochemical properties of amino acids	6
2.2	Encoding the protein sequences	7
2.2.1	Encoding methods using the amino acid sequence	7
2.2.2	Encodings using PCPs of amino acids	10
2.3	Machine Learning Techniques	18
2.3.1	Cluster Analysis	18
2.3.2	Classification	20
2.4	Feature selection and reduction techniques	27
2.4.1	Feature selection	27
2.4.2	Feature extraction	28
3	Literature Review	30
3.1	Importance of encoding protein sequences using the physico-chemical properties	30
3.2	Representing the amino acids based on PCPs	31

TABLE OF CONTENTS

3.2.1	Selection of native amino acids PCPs	31
3.2.2	Derivation of novel descriptors	32
3.3	Classification based on clustering	35
3.4	Thesis contribution	36
4	Data and Methods	38
4.1	Datasets	38
4.1.1	Membrane proteins benchmark	39
4.1.2	MHC class II benchmark	39
4.1.3	Caspase-3 benchmark	41
4.2	General description of the proposed approach	42
4.3	Native and derived descriptors used in this study	44
4.3.1	The native properties	44
4.3.2	Kidera's properties	46
4.3.3	Atchley's properties	47
4.3.4	Venkatarajan's properties	47
4.3.5	Maetschke's properties	48
4.3.6	Georgieva's properties	48
4.3.7	Georgieva's BLOSUM62 properties	49
4.4	Encoding protein sequences using PCPs	49
4.5	Classification based on clustering	52
4.5.1	Data division step	53
4.5.2	Clustering step	53
4.5.3	Distribution step	54
4.5.4	Classification step	55
4.6	Prediction of a new testing sample	56
4.7	Performance evaluation	57

TABLE OF CONTENTS

5 Experiments and Results	60
5.1 Experimental settings	60
5.2 Results from full protein sequences	61
5.3 Results from peptide sequences	64
5.3.1 Results from MHC-II sequences	64
5.3.2 Results from Caspase sequences	65
5.4 Time Performance	69
5.5 Discussion of results	72
5.5.1 Performance of encoding methods in our approach . . .	72
5.5.2 Performance of selected descriptors in our approach . .	74
5.5.3 Performance of our proposed approach on different bench- marks	75
5.5.4 Performance of SVM training time in our approach . .	76
6 Conclusion and Future Work	77
6.1 Mapping data into a higher dimensional feature space	78
6.2 Maximal margin separating hyperplanes	83
6.3 Maximum Margin	84
6.4 Example of maximum protein sequences	40
6.5 Example of MHC-II peptides	41
6.6 Example of Caspase-3 peptides	42
6.7 Block diagram of the proposed approach	43
6.8 Example of cross validation	54
6.9 Clustering step	55
6.10 Distribution step	56
6.11 The classification step	57
6.12 The four possible outcomes of the classifier	58

LIST OF FIGURES

List of Figures

2.1 Accuracy of SVM for membrane proteins using PseAAC. 62

2.2 Accuracy of SVM for membrane proteins using CTD. 63

2.3 Accuracy of SVM for MHC-II sequences using PseAAC. 65

2.4 Accuracy of SVM for MHC-II sequences using CTD. 66

2.5 Accuracy of SVM for Caspase sequences using PseAAC. 67

2.6 Accuracy of SVM for Caspase sequences using CTD. 68

2.7 Accuracy of SVM for Caspase sequences using concatenating 69

2.1 An example of AC encoding 8

2.2 An example of the dipeptide composition approach based on 9

2.3 An example of the OE encoding 9

2.4 An example of encoding using concatenating methods 10

2.5 An example of average physicochemical encoding. 11

2.6 An example of clustering showing intra and inter distances 19

2.7 Diagram of classification method. 21

2.8 Mapping data into a higher dimensional feature space. 22

2.9 Many possible separating hyperplanes 23

2.10 Maximum Margin 24

4.1 Example of membrane protein sequences. 40

4.2 Example of MHC-II peptides. 41

4.3 Example of Caspase-3 peptides. 42

4.4 Block diagram of the proposed approach. 43

4.5 Example of cross validation. 54

4.6 Clustering step. 55

4.7 Distribution step. 56

4.8 The classification step. 57

4.9 The four possible outcome of the classifier 59

LIST OF FIGURES

5.1	Accuracy of SVM for membrane proteins using PseACC.	62
5.2	Accuracy of SVM for membrane proteins using CTD.	63
5.3	Accuracy of SVM for MHC-II sequences using PseAAC.	65
5.4	Accuracy of SVM for MHC-II sequences using CTD.	66
5.5	Accuracy of SVM for Caspase sequences using PseAAC.	67
5.6	Accuracy of SVM for Caspase sequences using CTD.	68
5.7	Accuracy of SVM for Caspase sequences using concatenating method.	69
5.8	The time performance for the proposed approach based on SVM algorithm.	72
4.1	The description of the native properties	40
4.2	Values of Kider's factors	40
4.3	Values of Atchley's factors	47
4.4	Values of Varshadajan's factors	48
4.5	Values of Maciejko's Factors	49
4.6	Values of the Georgiev factors	50
4.7	Values of Georgiev-BLOSUM62 factors	51
5.1	Comparison the time performance and accuracy for different datasets	71

LIST OF FIGURES

5.1	Accuracy of SVM for membrane proteins using PseACC. . . .	62
5.2	Accuracy of SVM for membrane proteins using CTD.	63
5.3	Accuracy of SVM for MHC-II sequences using PseAAC.	65
5.4	Accuracy of SVM for MHC-II sequences using CTD.	66
5.5	Accuracy of SVM for Caspase sequences using PseAAC. . . .	67
5.6	Accuracy of SVM for Caspase sequences using CTD.	68
5.7	Accuracy of SVM for Caspase sequences using concatenating method.	69
5.8	The time performance for the proposed approach based on SVM algorithm.	72
4.1	The description of the native properties	40
4.2	Value of Kolar's factors	46
4.3	Value of Achley's factors	47
4.4	Value of Verbitskaya's factors	48
4.5	Value of Marchal's factors	49
4.6	Value of the Geary's factors	50
4.7	Value of Geary's-BLOSUM2 factors	51
5.1	Comparison the time performance and accuracy for different datasets	71

List of Tables

2.1	Examples of physicochemical properties	7
2.2	Distribution the amino acids into groups based on their PCPS	16
4.1	The description of the native properties	45
4.2	Values of Kidera's factors	46
4.3	Values of Atchley's factors	47
4.4	Values of Venkatarajan's factors	48
4.5	Values of Maetschke's Factors	49
4.6	Values of the Georgieve factors	50
4.7	Values of Georgieve BLOSUM62 factors	51
5.1	Comparison the time performance and accuracy for different datasets	71

List of Abbreviations

AA	Amino Acid
AC	Amino acid Composition
CTD	Composition, Transition and Distribution
FA	Factor Analysis
MDS	Multidimensional Scaling
MHC	Major Histocompatibility Complex
OE	Orthonormal Encoding
OMPs	Outer Membrane Proteins
PCA	Principle Component Analysis
PseAAC	Pseudo Amino Acid Composition
PCPs	Physicochemical Properties
SVM	Support Vector Machines

Chapter 1

Introduction

Proteins represent an important component in living cells. They perform most biological functions inside and outside them and determine the overall body status in health and disease [28]. Each protein within a given organism has a specific role. Without proteins, the organisms would be unable to reform, adjust or protect themselves [61].

In the field of Bioinformatics, prediction of biological attributes such as function, structure and localization, based on protein sequences is gaining more attention[53]. Using machine learning algorithms, predication is used to identify the family or the functional class to which a newly discovered protein belongs, and it helps the researcher to identify the functions and structures of unknown proteins in a faster, more accurate, and more cost effective manner [53].

Recently, several researchers have focused on using different classification techniques to solve various protein prediction problems, such as assigning function, structure, sub-cellular location, and role in interaction networks...etc.

In bioinformatics, proteins are represented as strings of characters of vari-

able lengths as follows: Let $s = r_1, r_2, \dots, r_n$ be a protein sequence of length $|s| = n$ over an alphabet Σ , where r_i represents the i_{th} residue in the sequence, and $\Sigma = \{G, A, V, L, I, P, F, Y, W, S, T, N, Q, C, M, D, E, H, K, R\}$. Each element in Σ is called amino acid. Usually when $n < 50$ we refer to the protein sequence as a *peptide*[39].

When we wish to apply machine learning techniques, such as classification or clustering, to protein sequences or peptides, we are faced with two facts. 1) the proteins are represented as characters and as not numeric values and 2) the proteins have different lengths. Since machine learning techniques usually require that all input data be numeric and fixed length, we need to encode the proteins into a new representation.

Formally an encoding method can be considered as a transform $x = E(s_i, p)$, where s_i is the protein sequence of arbitrary length n_i and x is the encoded vector of length p . This means that the encoding transform E unifies the length of the protein sequences to a given length p , which makes the classification process possible. The values in s are also changed by the encoding method accordingly with minimum loss of information.

The evaluation process of any classification method is usually performed by first dividing the data of interest into training and testing set. Then, the classifier is trained to map each element of the training set to a given class. After that, the classifier is evaluated by measuring its ability to correctly predict (classify) the elements of the testing set based on the gained knowledge through training.

A binary classifier is a special classifier which can recognize two classes. The formal definition of binary classification is as follows, we are given a training dataset, $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$, where x_i represents a high dimensional feature vector of a given length m and $y_i \in \{-1, 1\}$ is

1.1. THESIS OBJECTIVE

its corresponding label. In the case of protein prediction, x_i is an encoded protein sequence and y_i is its corresponding biological binary propriety. Now, Let τ be a classification method defined as $\tau(D) = \gamma$, where γ is the learnt experience through the classification function τ . Once we obtain γ , we can apply it to further classification of novel elements.

Usually classification leads to very good to excellent results, when the data of interest can be easily separable. This is not always the case. therefore sometimes we need to process the data prior to classification as explained below.

1.1 Thesis objective

In this thesis, we propose a method to enhance the performance of prediction (training time and accuracy)for protein attributes using a combination of classification algorithms and clustering analysis. We apply clustering prior to classification. A separate classifier for each cluster is used for protein prediction. This will make the classifiers work with easily separable data inside each cluster and will eventually enhance the prediction power of the classifiers.

1.2 Contributions

The following summarise the main contributions to the thesis:

- We propose a new approach that will enhance the prediction accuracy and computation time of protein attribute prediction by applying clustering prior or classification.
- We study the effect of different encoding methods on the performance

1.3. THESIS ORGANIZATION

of the proposed approach.

- We verify the results using different data sets in order to ensure general applicability of the approach regardless of the protein problem

1.3 Thesis Organization

The remaining parts of the thesis are organized as follows: chapter 2 describes the theories and basic concepts that are needed to understand the rest of the thesis. Chapter 3 contains a summary of some previous works related to our work. Chapter 4 covers the methodology used in this thesis to enhance the accuracy of the prediction, and the description of the benchmarks. Chapter 5 demonstrates experiments and the results achieved by the work, and results discussion. Finally, Chapter 6 concludes the work and propose some new direction for the future work.

ies [37]. In this thesis, we refer to these properties as native properties. In addition, researchers have generated new properties from the native properties which we refer to as derived properties (see Section 4.3).

Table 2.1 presents an example of 3 physicochemical properties of 10 amino acids: size, charge and hydrophobic (a measure of how strongly the side chains are pushed out of the water) [18].

Chapter 2

Background

Background of physicochemical properties. An example of three physicochemical properties (size, charge and Hydrophobic) for 10 amino acids, taken from the AAindex database.

Amino acid	R	K	D	Q	N	E	H	S	T	P
Size	136	129	115	128	114	129	17	87	101	87

This chapter gives a theoretic background needed for understanding the rest of the thesis. The first section explains the physicochemical properties of amino acids. The second section explains the encoding methods for protein sequences. In the third section of this chapter, the machine learning techniques needed in this thesis are explained, such as K-mean clustering algorithm and the SVM classifier. The final section covers the main techniques for feature selection and extraction.

2.1 Physicochemical properties of amino acids

Amino acids that form the proteins determine the properties of proteins, each amino acid has a set of physicochemical properties (PCPs), these PCPs can be used to study protein sequence profiles, folding and function [37].

The amino acid properties can be represented by the set of numerical values, which are known as the amino acid indices [57].

A few databases of amino acid indices have been constructed and regularly maintained. The most important ones are: AAindex and APDbase [30] [37]. AAindex contains 544 properties [30], where APDbase contains 242 proper-

2.2. ENCODING THE PROTEIN SEQUENCES

ties [37]. In this thesis, we refer to these properties as native properties. In addition, researchers have generated new properties from the native properties which we refer to as derived properties (see Section 4.3).

Table 2.1 presents an example of 3 physicochemical properties of 10 amino acids, the properties are size, charge and hydrophobic (a measure of how strongly the side chains are pushed out of the water) [18].

Table 2.1: Examples of physicochemical properties. An example of three physicochemical properties (size, charge and Hydrophobic) for 10 amino acids, these values are taken from AAindex database

Amino acid	R	K	D	Q	N	E	H	S	T	P
Size	156	128	115	128	114	129	137	87	101	97
Charge	10.8	9.7	2.8	5.7	5.4	3.2	7.6	5.7	5.9	6.5
Hydrophobic	-7.5	-4.5	-3	-2.9	-2.7	-2.6	-1.7	-1.1	-0.8	-0.3

2.2 Encoding the protein sequences

In order to apply machine learning algorithms to investigate protein sequences, the protein sequences need to be represented numerically. As defined previously in Section 1.1 the encoding is a transform function $X = E(s_i, p)$. The two major encoding methods of protein sequences are: encoding methods based on the amino acid sequence and encoding methods based on physicochemical properties of the amino acids [43]. This section consists description of some of these methods.

2.2.1 Encoding methods using the amino acid sequence

Different methods have been developed to encode the sequences using the amino acids characters. Some of these methods are:

2.2. ENCODING THE PROTEIN SEQUENCES

Amino acid composition (AC)

This is a simple encoding method. It finds the frequency of each amino acid in the protein sequence. Therefore, the encoded vector contains 20 numerical features regardless of the length of the protein sequence [10]. Figure 2.1 shows an example of AC encoding.

Peptide sequence **RQANFLGKIWP SHKGR**

Amino Acid	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
AC encoding	1	0	0	0	1	2	1	1	2	1	0	1	1	1	2	1	0	0	1	0

← Length = 20 →

Figure 2.1: An example of AC encoding

The dipeptide composition

The dipeptide is a component that contains two amino acids. Given that we have twenty amino acids, we can have a combinations of 400 dipeptides. The dipeptide composition calculates the frequency of each dipeptide in the sequence, this method has an advantage of taking into account the order of amino acids in the sequence [10]. Figure 2.2 shows an example of dipeptide composition encoding.

Orthonormal Encoding

Orthonormal Encoding (OE) is also called distributed encoding or sparse encoding. In OE, each amino acid is represented by a 20-bit vector with 19 bits set to zero and one bit set to one, the exist of amino acid at a given residue is encoded as 1 [43]. Figure 2.3 shows an example of OE encoding.

2.2. ENCODING THE PROTEIN SEQUENCES

2.2.2 Encodings using PCPs of amino acids

The PCPs of amino acids help to determine the structure and function of the protein sequence [34]. There are different methods used to encode sequences based on physicochemical properties of amino acids, some of these methods are :

Concatenating method

This is a simple method used to represent each amino acid numerically as a set of different physicochemical properties. For example if the length of the sequence is N and each amino acid represent by 5 properties, then the length of the feature vector will be $N \times 5$ [48]. So in this method, the length of the feature vector depends on the length of the protein sequence and the number of selected PCPs. Figure 2.4 shows an example of this encoding method.

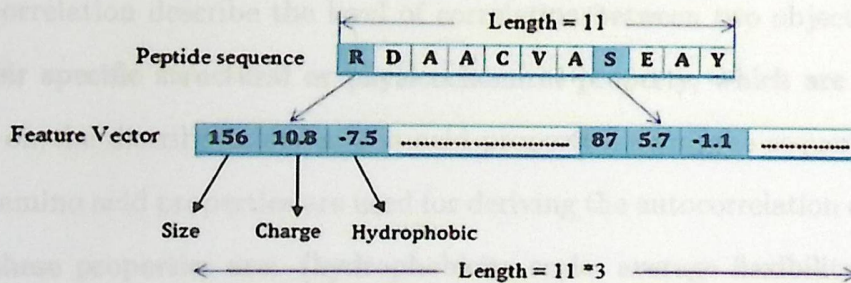


Figure 2.4: An example of encoding using concatenating methods, Three PCPs were used (size, charge, and hydrophobicity)

The average physicochemical encoding

This encoding is simple, and it invariant to the length of the sequence, thus mainly suited for proteins. Each feature is represented by the average value of a physicochemical property with respect to the amino acid in the sequence, therefore the feature vector is composed by F features where F is the number

2.2. ENCODING THE PROTEIN SEQUENCES

of selected PCPs [43]. Figure 2.5 shows an example of an average physicochemical encoding method.

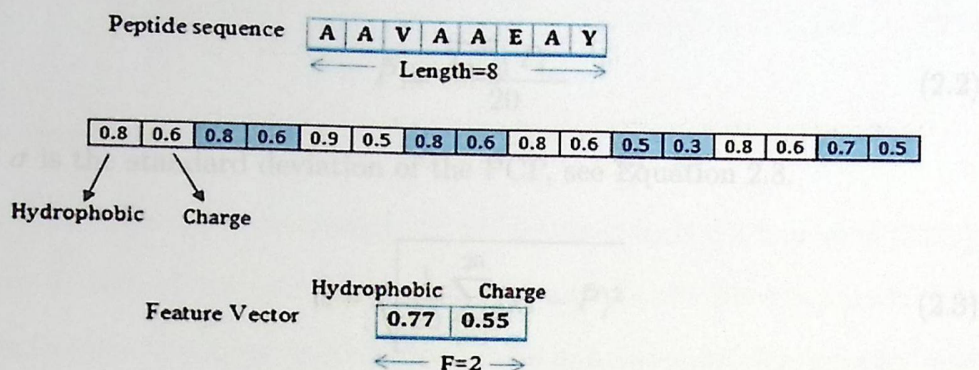


Figure 2.5: An example of average physicochemical encoding. Two PCPs were used (hydrophobic and charge) after normalized it to be between 0 and 1

Autocorrelation

Autocorrelation describe the level of correlation between two objects based on their specific structural or physicochemical property, which are defined based on the distribution of amino acid properties along the sequence [44]. Eight amino acid properties are used for deriving the autocorrelation descriptors, these properties are: (hydrophobicity scale, average flexibility index, polarizability parameter, free energy of amino acid solution in water, residue accessible surface areas, amino acid residue volumes steric parameters, and relative mutability [44].

There exist mainly three types of autocorrelation descriptors: Moreau-Broto, Moran and Geary autocorrelation descriptors. All PCPs values of amino acids should be normalized before applying these encodings, the normalization process describes in Equation 2.1.

$$P_n = \frac{P - \bar{P}}{\sigma} \quad (2.1)$$

2.2. ENCODING THE PROTEIN SEQUENCES

where P_n is the PCP after normalized, P is the PCP before normalized, \bar{P} is the mean of the PCP of the 20 amino acids and is defines in Equation 2.2.

$$\bar{P} = \frac{\sum_{i=1}^{20} P_i}{20} \quad (2.2)$$

and σ is the standard deviation of the PCP, see Equation 2.3.

$$\sigma = \sqrt{\frac{1}{20} \sum_{i=1}^{20} (P_i - \bar{P})^2} \quad (2.3)$$

1. Normalized Moreau-Broto autocorrelation descriptors

The normalized Moreau-Broto autocorrelation descriptors [44] can be defined as follows:

$$ATS(d) = \frac{\sum_{i=1}^{N-d} P_i P_{i+d}}{N-d} \quad d = 1, 2, 3, \dots, nlag \quad (2.4)$$

where:

- d is called the lag of the autocorrelation (e.g: lag 1 means correlating between the variable X_i and X_{i-1}).
- P_i and P_{i+d} are the properties of the amino acids at position i and $i + d$, respectively.
- $nlag$ is the maximum value of the lag.

2. Moran autocorrelation descriptors

The Moran autocorrelation descriptors [44] can be defined as follows:

$$I(d) = \frac{\frac{1}{N-d} \sum_{i=1}^{N-d} (P_i - \bar{P})(P_{i+d} - \bar{P})}{\frac{1}{N} \sum_{i=1}^N (P_i - \bar{P})^2} \quad d = 1, 2, 3, \dots, 30 \quad (2.5)$$

3. Geary autocorrelation descriptors

2.2. ENCODING THE PROTEIN SEQUENCES

The Geary autocorrelation descriptors [44] can be defined as:

$$I(d) = \frac{\frac{1}{2(N-d)} \sum_{i=1}^{N-d} (P_i - P_{i+d})^2}{\frac{1}{N-1} \sum_{i=1}^N (P_i - \bar{P})^2} \quad d = 1, 2, 3, \dots, 30 \quad (2.6)$$

The quasi-sequence-order descriptors

The quasi-sequence-order descriptors are proposed by K.C.Chou, et.al (2000). They are derived from the distance matrix between the 20 amino acids. The physicochemical properties computed include hydrophobicity, polarity, and side-chain volume [14].

1. Sequence-order-coupling Number The d -th rank sequence-order-coupling number is defined as [14]:

$$T_d = \sum_{i=1}^{N-d} (d_{i,i+d})^2 \quad d = 1, 2, 3, \dots, nlag \quad (2.7)$$

where $d_{i,i+d}$ is the distance between the two amino acids at position i and $i + d$.

2. Quasi-sequence-order Descriptors: In this case, for each amino acid type a quasi-sequence-order descriptor can be defined as [14]:

$$X_r = \frac{f_r}{\sum_{r=1}^{20} f_r + w \sum_{d=1}^{nlag} T_d} \quad r = 1, 2, 3, \dots, 20 \quad (2.8)$$

where f_r is the normalized occurrence for amino acid, and w is a weighting factor (often $w = 0.1$).

These are the first 20 quasi-sequence-order descriptors. The other

2.2. ENCODING THE PROTEIN SEQUENCES

quasi-sequence-order descriptors are defined as:

$$X_d = \frac{wT_{d-20}}{\sum_{r=1}^{20} f_r + w \sum_{d=1}^{nlag} T_d} \quad d = 21, 22, \dots, 20 + nlag \quad (2.9)$$

The pseudo amino acid composition

The pseudo amino acid composition (PseAAC) is similar to the quasi-sequence order descriptor, it proposed by Chou (2001) [15]. The pseudo amino acid descriptor is made up of a $(20+k)$ vector in which the first 20 components reflect the effect of the amino acid composition and the remaining components reflect the effect of sequence order by the correlation factors of the different ranks. The last K features are obtained based on a given physicochemical property [15].

The PseAAC can be described as follow:

If the protein sequence have L amino acids residues: $R_1R_2R_3\dots R_{L-2}R_{L-1}R_L$

Sequence order effect can be approximately reflected with a set of sequence order-correlated factors as defined below:

$$\begin{aligned} \theta_1 &= \frac{1}{L-1} \sum_{i=1}^{L-1} \Theta(R_i, R_{i+1}) \\ \theta_2 &= \frac{1}{L-2} \sum_{i=1}^{L-2} \Theta(R_i, R_{i+2}) \\ \theta_3 &= \frac{1}{L-3} \sum_{i=1}^{L-3} \Theta(R_i, R_{i+3}) \\ &\vdots \\ \theta_\lambda &= \frac{1}{L-\lambda} \sum_{i=1}^{L-\lambda} \Theta(R_i, R_{i+\lambda}) \quad (\lambda < L) \end{aligned} \quad (2.10)$$

The θ_1 is called the first-tier correlation factor that reflects the sequence order correlation between all the most contiguous residues along a protein chain, θ_2 the second-tier correlation factor that reflects the sequence order correlation between all the second most contiguous residues, and θ_λ is the λ -th tier correlation factor [15].

2.2. ENCODING THE PROTEIN SEQUENCES

The correlation factor can be defined as:

$$\Theta (R_i, R_j) = [F(R_j) - F(R_i)]^2 \quad (2.11)$$

where $F(R_i)$ is the feature (e.g. size) value of the amino acid R_i . The value is converted from the original feature value of the amino acid according to the following equation:

$$F(R_i) = \frac{F_0(R_i) - \sum_{i=1}^{20} \frac{F_0(R_i)}{20}}{\sqrt{\frac{\sum_{i=1}^{20} [F_0(R_i) - \sum_{i=1}^{20} \frac{F_0(R_i)}{20}]^2}{20}}} \quad (2.12)$$

where $F_0(R_i)$ is the original feature value of the amino acid R_i . So, the feature vector (V) of the protein can be represented by a $(20 + \lambda)$ vector as follows:

$$v_x = \begin{cases} \frac{f_x}{\sum_{i=1}^{20} f_{i+w} \sum_{j=1}^{\lambda} \theta_j} & (1 \leq x \leq 20) \\ \frac{W\theta_{x-20}}{\sum_{i=1}^{20} f_{i+w} \sum_{j=1}^{\lambda} \theta_j} & (21 \leq x \leq 20 + \lambda) \end{cases} \quad (2.13)$$

where $f_x (x = 1, 2, \dots, 20)$ represents the amino acid composition (AC), which was described earlier.

Composition, Transition and Distribution(CTD)

This method depending on distributing amino acids into groups based on their PCPs, it was developed by Dubchak et al. (1995). In this method the amino acids are divided into three classes according to its attribute and each amino acid is encoded by one of the indices 1, 2 and 3 according to which class it belonged. The amino acids distributed into three classes based on 7 physicochemical properties [17], see Table 2.2.

2.2. ENCODING THE PROTEIN SEQUENCES

Table 2.2: Distribution the amino acids into groups based on their PCPS [17]

Attributes	Group 1	Group 2	Group 3
Hydrophobicity	Polar (R,K,E,D,Q,N)	Neutral (G,A,S,T,P,H,Y)	Hydrophobicity (C,L,V,I,M,F,W)
Normalized van der Waals Volume	0-2.78 (G,A,S,T,P,D,C)	2.95-4.0 (N,V,E,Q,I,L)	4.03-8.08 (M,H,K,F,R,Y,W)
Polarity	4.9-6.2 (L,I,F,W,C,M,V,Y)	8.0-9.2 (P,A,T,G,S)	10.4-13.0 (H,Q,R,K,N,E,D)
Polarizability	0-1.08 (G,A,S,D,T)	0.128-0.186 (C,P,N,V,E,Q,I,L)	0.219-0.409 (K,M,H,F,R,Y,W)
Charge	Positive (K,R)	Neutral (A,N,C,Q,G,H,I,L, M,F,P,S,T,W,Y,V)	Negative (D,E)
Secondary Structure	Helix (E,A,L,M,Q,K,R,H)	Strand (V,I,Y,C,W,F,T)	Coil (G,N,P,S,D)
Solvent Accessibility	Buried (A,L,F,C,G,I,V,W)	Exposed (R,K,Q,E,N,D)	Intermediate (M,S,P,T,H,Y)

Each sequence converted into a new sequence where each amino acid is represented by a number of a group depended on each previous attribute. Then, we can find three values for each sequence, these values represents the composition (C), transition (T) and distribution (D).

Example: For a sequence: FAKITAAMCQEIDESSGHGA and according to the hydrophobicity division in Table2.2, the sequence is encoded as: 32132223311311222222

1. Composition: Composition can be defined as:

$$C_i = \frac{n_i}{N} \quad i = 1, 2, 3 \quad (2.14)$$

where n_i is the number of i in the encoded sequence and N is the length

2.2. ENCODING THE PROTEIN SEQUENCES

of the sequence [41], for each sequence we can find 21 values represent the composition for 7 attributes, and for each attributes three groups.

Based on the previous example, the composition values of the sequence are: $C_1 = 5/20$, $C_2 = 10/20$ and $C_3 = 5/20$ Where 20 is the length of the protein sequence.

2. Transition: The transition represent the transition from one group to another for the same attribute [41], e.g.: transition from class 1 to 2 is the percent frequency with which 1 is followed by 2 or 2 is followed by 1 in the encoded sequence. The transition can be defined as:

$$T_{ij} = \frac{n_{ij} + n_{ji}}{N - 1} \quad ij = [1, 2], [1, 3], [2, 3] \quad (2.15)$$

Also, for each sequence we can find 21 values represent the transition. Based on the previous example the transition values of the sequence are: $T_{12} = 2/19$, $T_{23} = 3/19$ and $T_{13} = 4/19$

3. Distribution: The distribution descriptor describes the distribution of each attribute in the sequence. There are five distribution descriptors for each group and they are the position percents in the sequence for the first residue, 25% residues, 50% residues, 75% residues and 100% residues [41]. For each sequence we can find 105 values represent the distribution for 7 attributes, and for each attributes three groups, where 5 values (residues) for each group.

Based on the previous example the distribution values of the group 2 in

2.2. ENCODING THE PROTEIN SEQUENCES

of the sequence [41], for each sequence we can find 21 values represent the composition for 7 attributes, and for each attributes three groups.

Based on the previous example, the composition values of the sequence are: $C_1 = 5/20$, $C_2 = 10/20$ and $C_3 = 5/20$ Where 20 is the length of the protein sequence.

2. Transition: The transition represent the transition from one group to another for the same attribute [41], e.g.: transition from class 1 to 2 is the percent frequency with which 1 is followed by 2 or 2 is followed by 1 in the encoded sequence. The transition can be defined as:

$$T_{ij} = \frac{n_{ij} + n_{ji}}{N - 1} \quad ij = [1, 2], [1, 3], [2, 3]$$

Also, for each sequence we can find 21 values represent the transition

Based on the previous example the transition values of the sequence

are: $T_{12} = 2/19$, $T_{23} = 3/19$ and $T_{13} = 4/19$

3. Distribution: The distribution descriptor describes the distribution of each attribute in the sequence. There are three distribution descriptors for each group and they are the position of the first residue, the percentage of the first residue, 25% residues, 50% residues and 75% residues [41]. For each sequence the distribution descriptor describes the distribution for 7 attributes and for each attribute three groups 5 values (residues) for each attribute. The distribution descriptors are substantial Based on the previous example the distribution descriptors for each single group are

2.3. MACHINE LEARNING TECHNIQUES

the sequence are: There are 10 amino acids encoded as 2 in the above example, the residues for the group 2 in the encoded sequence are 2 (for the first position), 5 (for 25% from the 10), 15 (for 50% from the 10), 17 (for 75% from the 10) and 20 (for 100% from the 10), so the distribution descriptors for group 2 are: 10.0 ($2/20 \times 100$), 25.0 ($5/20 \times 100$), 75.0 ($15/20 \times 100$), 85.0 ($17/20 \times 100$) and 100.0 ($20/20 \times 100$), respectively.

2.3 Machine Learning Techniques

Machine learning is concerned with the development of algorithms and techniques that allow computers to learn, it can be defined as a science of algorithmic methods of learning from experience with the goal of improving performance on selected tasks [40].

Mainly there are two types of machine learning, these types are [25]:

- Supervised learning: where both input and target pairs should be provided during the learning process, such as classification
- Unsupervised learning: where only input and no target is required during learning, such as clustering

This section introduces a description of classification and clustering.

2.3.1 Cluster Analysis

Clustering is a very common technique in unsupervised machine learning to discover groups of data that behave similarly based on features describes the objects. The result of cluster analysis is a number of heterogeneous groups with homogeneous contents inside each group, where there are substantial differences between the groups, but the individuals within a single group are

2.3. MACHINE LEARNING TECHNIQUES

similar [56]. One advantage of the clustering is that it can be used to reduce the data, by replacing all of the elements in a cluster with a single representative element. Formally, the aim of clustering is to automatically collect the data into groups (clusters) based on their similarities. A clustering algorithms re-arrange a dataset x_1, x_2, \dots, x_n into the clusters $\{c_1, c_2, \dots, c_k\}$, where $k < n$, such that the elements x_i and $x_j \in c_k$, iff $\Delta(x_i, x_j) < \epsilon$. otherwise x_i and x_j belong to different clusters, where Δ is a distance function and ϵ is predefined distance.

A good clustering method will produce high quality clusters in which the similarity in the intra-class is high, and the inter-class is low, see Figure 2.6

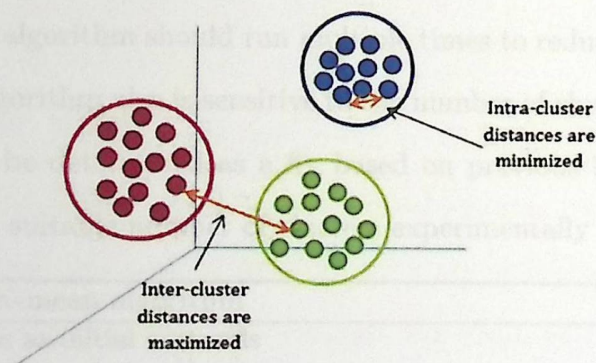


Figure 2.6: An example of clustering showing intra and inter distances. The distances between the instances within the same cluster should be minimized, and between the clusters should be maximized

There are different clustering approach such as K-mean [56], hierarchical clustering [56] and SOM clustering[60]. Here we explain the most common and simplest one, which is the K-mean.

K-mean

K-means algorithm is one of the simplest clustering algorithms that solve the well-known clustering problem. The k-mean algorithm classify a given data

2.3. MACHINE LEARNING TECHNIQUES

based on a certain number of clusters (assume k clusters), for each cluster the centroid should be defined [56]. K-mean is described by Algorithm 1.

First, the number of clusters and the initial centroids (points representing the centers of the clusters) should be determined, then each point in the sample assigned to the nearest cluster centroid using the Euclidean distance in Equation 2.16, where n represents the dimension.

$$\Delta = \left(\sum_{i=1}^n (x_i - y_i)^2 \right)^{1/2} \quad (2.16)$$

The cluster centroids are updated based on the mean of the data points in its cluster, the algorithm stops when the centroids do not change.

The K-mean algorithm is sensitive to the initial selected cluster centroids, so the k-means algorithm should run multiple times to reduce this effect [25]. The K-mean algorithm also is sensitive to the number of clusters, the number of clusters can be determined as a fix based on previous knowledge, or by trying to find a suitable number of clusters experimentally [25].

Algorithm 1 K-mean algorithm

```
select  $K$  points as initial centroids
for all centroid not change do
    Determine the distance of each object to the centroids.
    Group the object based on minimum distance.
    Recompute the centroids of new clusters.
end for
```

2.3.2 Classification

Classification is a very common technique in supervised machine learning to generate input-output mapping relations from a set of labeled training data [40]. Figure 2.7 illustrates the concept of classification.

Different machine learning methods can be used to solve classification

2.3. MACHINE LEARNING TECHNIQUES

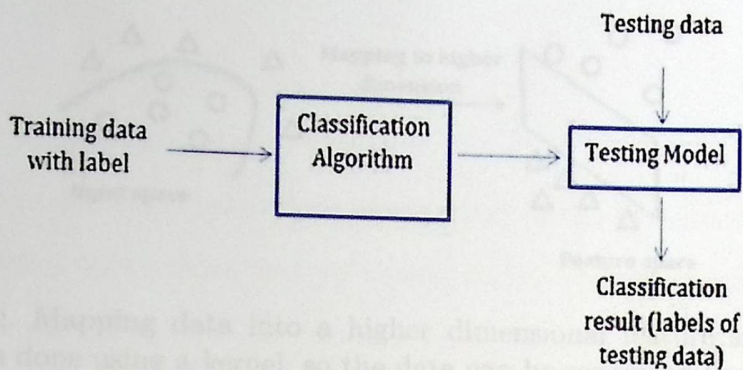


Figure 2.7: Diagram of classification method. The training data is inserted into the classification algorithm and after the training is done, the system can predicate the label of the testing data.

problem. In the following, support vector machine is explained as an example of classification tool.

Support Vector Machine

Support Vector Machine (SVM) is a supervised learning technique that generates input-output mapping relations from a set of labeled training data. SVM is a linear classifier that can separate the data, so that it can maximize the margin defined (maximizes the distance between it and the nearest data point of each class); the result is a hyperplane that separate the two classes. The SVM can be applied for classification and regression [23]. In this subsection the SVM for classification will be described.

To use the SVM, the input data should be transformed into a high-dimensional feature space using the nonlinear kernel functions. In order to make input data more separable [23]. Figure 2.8 illustrates the mapping to higher dimensional space.

SVM is a binary classier. The data for a two class learning problem consists of objects labeled with one of the two labels corresponding to the two

2.3. MACHINE LEARNING TECHNIQUES

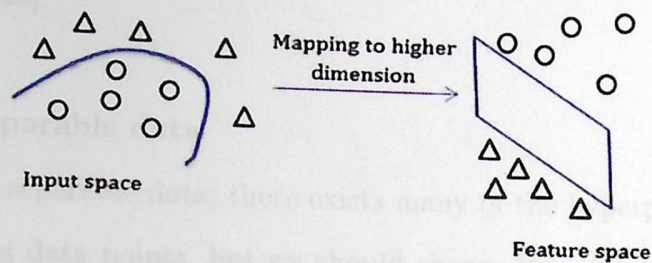


Figure 2.8: Mapping data into a higher dimensional feature space. The mapping is done using a kernel, so the data can be separated linearly

classes; for suitability we assume the labels are $+1$ (positive examples) or -1 (negative examples) [23].

Let L is a training points, where each input x_i has D attributes ($D - dimensions$) and is in one of two classes $y_i = -1$ or $y_i = +1$. In general the **linear classifier** can be defined as the dot product between two vectors, as follows:

$$\langle w, x \rangle = \sum_{j=1}^M w_j x_j \quad (2.17)$$

A **linear classifier** is based on a linear discriminant function of the form

$$f(x) = \langle w, x \rangle + b \quad (2.18)$$

where w is weight vector, and b is the bias, $f(x)$ assigns score for each point (x) in order to classify the point according to this score.

The hyperplane can be described by $f(x) = 0$, this hyperplane divides the space into two half spaces according to the sign of $f(x)$, that indicates on which side of the hyperplane a point is located, if $f(x) > 0$, then one decides for the positive class, otherwise for the negative. The boundary between regions classified as positive and negative is called the decision boundary of

the classifier [23].

Linear separable data

For the linear separable data, there exists many of the hyperplane that correctly classifies data points, but we should choose the optimal hyperplane, that maximizes the margin [9]. Figure 2.9 illustrates the possible separating hyperplanes for a set of data.

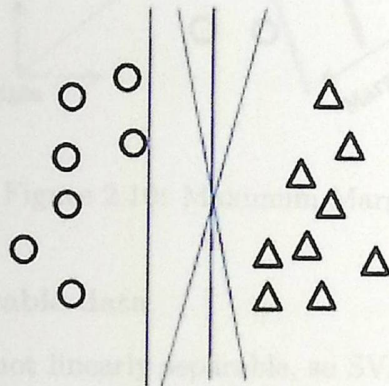


Figure 2.9: Many possible separating hyperplanes

To find the optimal hyperplanes all points should confirm the following constraint

$$y_i [\langle w, x_i \rangle + b] \geq 1 \quad \forall i = 1, 2, \dots, n \quad (2.19)$$

Also we should find the optimal b and w corresponding to the maximum margin hyperplane; one has to solve the following optimization problem [9].

$$\underset{w, b}{\text{minimize}} \frac{\|w\|^2}{2} \quad (2.20)$$

where the minimizing process in the previous equation means maximizing the margin.

The classifier that is applicable to the linearly separable data is called a hard margin SVM [9]. See Figure 2.10.

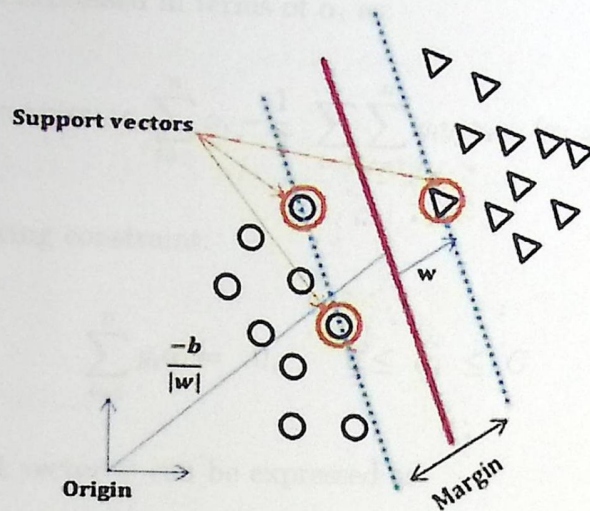


Figure 2.10: Maximum Margin

Non-linear separable data

In, practice, data are not linearly separable, so SVM provides a soft margin SVM for this type of data, that provides a greater margin that allows the classifier to mis-classify some data, by allowing errors, so the constraint on points will be changed to the following [9].

$$y_i [\langle w, x_i \rangle + b] \geq 1 - \xi_i \quad \forall i = 1, 2, \dots, n \quad (2.21)$$

where $\xi \geq 0$ are slack variables that allow data to be in the margin or misclassified, and, the optimization problem will be as follows [9].

$$\text{minimize}_{w,b} \frac{\|w\|^2}{2} + C \sum_{i=1}^n \xi_i \quad (2.22)$$

The constant $C > 0$ sets the relative importance of maximizing the margin and minimizing the amount of slack [9].

To solve the previous optimization problem the method of Lagrange mul-

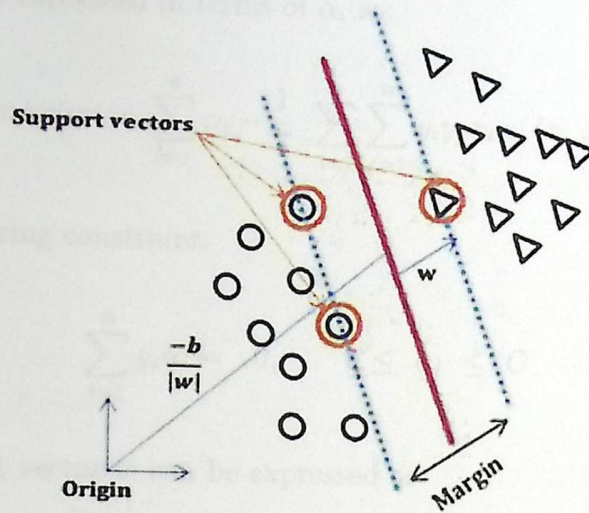


Figure 2.10: Maximum Margin

Non-linear separable data

In, practice, data are not linearly separable, so SVM provides a soft margin SVM for this type of data, that provides a greater margin that allows the classifier to mis-classify some data, by allowing errors, so the constraint on points will be changed to the following [9].

$$y_i [\langle w, x_i \rangle + b] \geq 1 - \xi_i \quad \forall i = 1, 2, \dots, n \quad (2.21)$$

where $\xi \geq 0$ are slack variables that allow data to be in the margin or misclassified, and, the optimization problem will be as follows [9].

$$\text{minimize}_{w,b} \frac{\|w\|^2}{2} + C \sum_{i=1}^n \xi_i \quad (2.22)$$

The constant $C > 0$ sets the relative importance of maximizing the margin and minimizing the amount of slack [9].

To solve the previous optimization problem the method of Lagrange mul-

2.3. MACHINE LEARNING TECHNIQUES

multipliers are used, it reformulates the original primary problem into dual formalization; it is expressed in terms of α_i as:

$$\text{maximize}_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle \quad (2.23)$$

under the following constraint:

$$\sum_{i=1}^n y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C \quad (2.24)$$

Then the weight vector w can be expressed as

$$w = \sum_{i=1}^n y_i \alpha_i x_i \quad (2.25)$$

The x_i , for which $\alpha_i > 0$ are called support vectors, see Figure 2.10.

The data that relate to non-linearly separable should be mapped to higher vector space using the mapping function (ϕ), then the discriminant function expressed as [9].

$$f(x) = \langle w, \phi(x) \rangle + b \quad (2.26)$$

In Equation 2.26 $f(x)$ is linear function that because it defined using the mapping function.

The mapping can be done using kernels, the weighting vector w is updated using the kernel as follows [9].

$$w = \sum_{i=1}^n y_i \alpha_i \phi(x_i) \quad (2.27)$$

2.3. MACHINE LEARNING TECHNIQUES

Then the new w substituting in the discriminant function, as follows

$$f(x) = \sum_{i=1}^n y_i \alpha_i \langle \phi(x), \phi(x_i) \rangle + b \quad (2.28)$$

where the $\langle \phi(x), \phi(x_i) \rangle$ is a kernel function, that defined as follows [9].

$$k(x, x_i) = \langle \phi(x_i), \phi(x) \rangle \quad (2.29)$$

Kernel Functions

Different kernel functions can be used with SVM, the common kernel functions are [8]:

1. Linear kernel: it is the simplest kernel function. It is computed by the inner product plus an optional constant as follows

$$k(x, y) = x^T y + C \quad (2.30)$$

2. Polynomial kernel: it is suitable for problems where all the training data is normalized.

$$k(x, y) = (\alpha x^T y + C)^d \quad (2.31)$$

where α is the slope that is an adjustable parameter and d is the degree of the polynomial.

3. Gaussian kernel: it is an example of a radial basis function kernel.

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) \quad (2.32)$$

where $\sigma > 0$ is a parameter that controls the width of the Gaussian, it

2.4. FEATURE SELECTION AND REDUCTION TECHNIQUES

plays a similar role as the degree of the polynomial kernel.

2.4 Feature selection and reduction techniques

When data objects that will be used by machine learning techniques are described by a large number of features (i.e. The data is high dimension), it is often beneficial to reduce the dimension of the data [16, 20].

Dimensionality reduction is the transformation of high-dimensional data into a meaningful representation of reduced dimensionality [35].

Dimensionality reduction is an important task in machine learning for different reasons [3] as follows:

- Facilitates classification, compression, and visualization of high-dimensional data.
- When an input is unnecessary (e.g. redundant), we save the cost of extracting it.
- Reduced both the time and space complexity.

After using a dimensional reduction techniques some information will be lost, but this information is considered the less important and have a weak ability to represent the data (unimportant features). There are two main methods for reducing dimensionality: feature selection and feature extraction.

2.4.1 Feature selection

In feature selection, a set of D dimensions that give us the most information is selected and the other dimensions (unimportant features) are discarded. There are two approaches for feature selection: forward and backward selection [3].

2.4. FEATURE SELECTION AND REDUCTION TECHNIQUES

In forward selection, we start with an empty set and add features one by one, at each step adding the one that decreases the error to the most, until any further addition does not decrease the error. In backward selection, we start with all features and remove them one by one, at each step removing the one that decreases the error the most (or increases it only slightly), until any further removal increases the error significantly [3].

2.4.2 Feature extraction

In feature extraction, a new set of k dimensions that are extracted from the original D dimensions is generated. These methods may be supervised or unsupervised depending on whether or not they use the output information [3].

Three methods of feature extraction are discussed below, these methods are: Principal Components Analysis, Factor Analysis and Multidimensional Scaling.

Principle Component Analysis

Principle Component Analysis (PCA) is the dimension reduction technique that is widely used in many applications due to its simplicity and efficiency [12]. The PCA can be calculated as follows: Let $\{D = x_i\}_{(i=1)}^n$ is a sample data described by a set of p features. This data can be represented by a feature-object matrix $X_{[p \times n]}$, where each column represents an object, the covariance of these data defined as [12]

$$C = \frac{XX^T}{n-1} \quad (2.33)$$

where the diagonal terms in C capture the variances in the individual features and the off-diagonal terms quantify the covariances between the

2.4. FEATURE SELECTION AND REDUCTION TECHNIQUES

corresponding pairs of features [12].

For this covariance matrix C , the eigen-vectors V and eigen-values Λ can be calculated. There exist p eigenvalues and eigen-vectors. The process of dimension reduction is started by selecting k eigen-vector (where $k < p$) with the highest eigenvalues of the data [3]. This produce a subset of Λ that we denote $\hat{\Lambda}$. Finally, the reduced data $\Phi_{[k \times n]}$, which has only k features can be computed by the following transformation:

$$\Phi = \hat{\Lambda} \times X^T \quad (2.34)$$

Factor Analysis

Factor Analysis (FA) is used when there exist a group of variables that have high correlation among themselves and low correlation with all the other variables, then there may be a single underlying factor that gave rise to these variables [3]. FA depends on partitioning the features into factor clusters, and then few factors can represent these groups of features. In contrast with PCA, in FA we can obtain the original features from the factors but in PCA we can not [3].

Multidimensional Scaling

Multidimensional Scaling (MDS) can be used when the distances between the pairs of points $d_{ij} \forall i, j = 1, 2, \dots, N$ are known, but the exact coordinates of the points, their dimensionality, or how the distances are calculated are unknown [3]. MDS is the method for placing these points in a low dimensional space where the distance between them is as close as possible to the given distances in the original space $d(i, j)$ [3].

Chapter 3

Literature Review

This chapter contains a summary of some important contributions related to our work. The chapter includes different approaches developed to select a suitable PCPs for amino acids, the importance of encoding protein sequences using the amino acids PCPS, and the earlier work related to use the clustering with classification to increase the prediction accuracy. The last section explains our contribution in this thesis.

3.1 Importance of encoding protein sequences using the physicochemical properties

The representation of the protein sequences using the physicochemical properties is very useful for machine learning prediction of protein structural and functional classes, protein-protein interactions and subcellular locations [26].

Many researches have showed that using a few important PCPs to encode the sequences is better than using the amino acid characters, and can improve the result of protein prediction.

In their work, Ray et al. applied different PCPs in order to predict

3.2. REPRESENTING THE AMINO ACIDS BASED ON PCPS

the peptide-MHC class I binding. In their results they found that using an important physicochemical features gave better results than using amino acids characters, whether these properties used separately or combined using different machine learning algorithms [51].

3.2 Representing the amino acids based on PCPs

As previously mentioned, a few databases of amino acid indices have been constructed and regularly maintained, these databases contain hundreds of amino acid properties, some of these properties are related to each other reflecting a high degree of redundancy. Several approaches have been followed to select a suitable subset in order to reduce the redundancy among the different properties, and to reduce the dimensionality of feature vectors. This subset should represent the main important properties that can be used to solve specific problems or for general use.

As mentioned in Chapter 2, there are two methods used to reduce the dimensionality of feature vectors; feature selection and feature extraction. The feature that were used to represent the amino acids using the amino acids PCPs can be divided into two groups; the first group represents the features selected from the databases of amino acid indices, the second group represents the features derived mathematically from the databases of amino acid indices.

3.2.1 Selection of native amino acids PCPs

In this approach the feature selection approach was used to reach an optimal subset of features [62, 52]. The selection process is done on the databases of

3.2. REPRESENTING THE AMINO ACIDS BASED ON PCPS

amino acid indices, so the values they used are real values coming from these databases.

Ray, et al. presented a selection approach depending on a set of properties based on the literature survey to predict peptide-MHC class-I binding. For any particular classifier, they started with the initial set of properties and employ the forward selection method using the mis-classification error as the criterion to choose a subset. On the other hand, Xiong, et al. started from AAindex database after removing the indices with missing values, and then they followed a similar approach with, by developing a greedy approach in combination with correlation analysis for feature selection, the final subset contains a four physicochemical properties. These two subsets of features [62, 52] is not for general use, but they are suitable for the specific problems.

3.2.2 Derivation of novel descriptors

The second approach is to derive a new subset of features by performing a reduction algorithm on the databases or on the amino acid substitution matrices, some of these approaches depending on different algorithms of reduction such as principal component analysis (PCA), Multidimensional Scaling (MDS) and Factor Analysis (FA), these approaches are useful when the purpose of the analysis is dimensionality reduction, but they are less useful in designing interpretable scales [22].

The derived methods started few decade ago, when Sneath [55], Kidera et al. [32] and Hellberg et al. [24] developed the approaches that aim at reducing the redundancy. These approaches could not help to solve the problems that have appeared recently, because it did not take into account the structural features (properties) of amino acids [22]. For example Kidera performed the FA on all available sets of physical properties of the 20 amino

3.2. REPRESENTING THE AMINO ACIDS BASED ON PCPS

acids. They demonstrated that all of these data can be represented by a set of 10 property factors. These factors correlated with α -helical propensity, bulk, β -sheet propensity, and hydrophobicity [32].

Recently, a new approaches have been developed that take into account different aspects of physicochemical properties of amino acids to derive amino acid descriptor scales.

Sandberg et al. derived five descriptors (z1-z5) using PCA algorithm on 26 different PCPs. These descriptors represent essentially hydrophobicity/hydrophilicity (z1), steric/ bulk properties and polarizability (z2), polarity (z3), and electronic effects (z4 and z5) of the amino acids [54].

However, Opiyo and Moriyama noticed that the z-scales derived by Sandberg et al. (1998) gave poor results for their classification problem because they lack structure related features. Opiyo and Moriyama applied PCA on 12 selected physicochemical properties (mass, volume, surface area, hydrophilicity, hydrophobicity, isoelectric point, transfer of energy solvent to water, refractivity, nonpolar surface area, the frequencies of α -helix, β -sheet, and reverse turn), then the first five principal components (PCPs) were selected [45].

The previous methods (Sandberg et al., Opiyo and Moriyama) were designed for a specific problem (GPCRs classification). On the other hand; Venkatarajan and Braun derived new 5 quantitative descriptors based on MDS of 237 physicochemical properties and they designed it for general use, these 5 descriptors correlate well with five properties (hydrophobicity, size, preferences for amino acids to occur in α -helices, number of degenerate triplet codons, and the frequency of occurrence of amino acid residues in β -strands) [59]. Also, Atchley et al. developed an approach for general use by applying factor analysis followed by promax rotation, in order to compute five

3.2. REPRESENTING THE AMINO ACIDS BASED ON PCPS

factors from 54 selected amino acid attributes. The promax rotation is used to find the simple structure in the data and for improving the interpretability of principal components [5].

In his study, Georgiev explained that Atchley et al., 2005 method gave a poor interpretability of two of the resulting five scales, because they used a small subset of properties during the analysis. Georgiev, 2009 work proved that a reduced dataset with lower redundancy could not be represented sufficiently well by less than 12 independent principal components. Therefore, Georgiev derived a 19 descriptors from 509 amino acid indices using the varimax criterion rather than the PCA to increase the ease of interpretation, also varimax scales gave a better performance than Atchley et al., 2005 in the task of Class A GPCR subfamily classification [22].

In his work, Georgiev performed another approach that depends on deriving new features from substitution matrix (technique used to find the similarity between sequences, but this technique depends on the PCPs of amino acids in order to determine how the amino acids substitute one another [29, 31]). He derived 10 factors from BLOSUM 62 substitution matrix, and he found that these factors gave a better result than all previous scales in the task of Class A GPCR subfamily classification. The result of Georgiev (2009) work demonstrated that the varimax scales are suitable for exploratory analyses, while the BLOSUM 62 scales appear better choice for unsupervised learning and modeling applications [22].

In an earlier work, Maetschke et al. derived a 5 factor from BLOSUM 62 to encode the peptide sequence in order to improve the single peptide cleavage site prediction, these 5 factors improved the result of this problem compared to the previous methods were used [36].

3.3 Classification based on clustering

A lot of efforts have been directed towards using an unsupervised with supervised machine learning techniques for two main purposes:

- Minimizing the computational time and memory consumption.
- Enhancing the accuracy of prediction.

Classification based on clustering has been used for different types of data, such as: text data, large numerical dataset, waveform data and others.

Cervantes, et al. and Yu, et al. have introduced approaches to reduce the classification time for large data set. In their work Cervantes, et al. generated a large random data set, they used a fuzzy clustering algorithm to cluster the training data, and then they kept a heterogeneous clusters (clusters contain data from different classes) for next steps and applied SVM on homogeneous clusters to find the optimal hyperplane, then they eliminated the homogeneous clusters far away from the optimal hyperplane, after that, the de-clustering and the SVM classification via reduced data were used, so using this method enabled them to reduce the training time while maintain the same range of accuracy [13]. Their approach as we can see eliminates some samples from the dataset, and also it adds an overhead for applying the SVM classifier twice.

Yu, et al. approach is similar to previous but they used a hierarchal clustering rather than the fuzzy, and they were able to enhance the time of classification for lager dataset, but they showed that random sampling could hurt the training process of SVM, especially when the probability distribution of training and testing data were different [63].

Kyriakopoulou, et al. have enhanced the classification result for text data by clustering the data into clusters and then each cluster contributes

3.4. THESIS CONTRIBUTION

one meta-feature to the feature space of the training and testing data, finally they used SVM classifier to classify the expanded data (data contains the original features and meta features), they were able to enhance the classifier results approximately by 8% [33]. The main disadvantage of this method is that the testing data should be involved in the process from the beginning to form the meta-features.

Rahideh, et al. have studied the cancer data (colon cancer and leukemia) by using the clustering in order to group the genes and then select the top ranking genes from each group to form the intended sub-set of relevant genes to be used for classification. As a result, they found that the accuracy of the classifiers with and without clustering is comparable for the cancer sequences [49].

3.4 Thesis contribution

Reviewing previous literature showed that there are few works using the clustering for classification that were mainly focusing on reducing the complexity of classification for specific types of data such as random numerical data or text data. On the other hand, few other studies have focused on using the clustering before the classification to improve the classification accuracy.

As we see from the previous works and up to our knowledge, no attempts had been made to study the importance of clustering the protein sequences data before the classification in order to improve the classification performance.

This thesis is concerned with improving the classification performance (computational time and accuracy) for the protein sequences using the machine learning algorithms.

3.4. THESIS CONTRIBUTION

Our approach aims at exploring the importance of clustering protein sequences into groups based on their similarities, then applying the classification on each group rather than applying the classification of all the data set in order to enhance the performance of prediction.

Our novel method depending on groups the training data using clustering algorithm and distribute the testing data into these groups based on the distances between the test data and the centroids of clusters, then applying the classification algorithm on these clusters. The most important features that distinguish our approach from previous approaches are: there is no need to eliminate samples from the dataset in order to minimize the computation time as the previous approaches, and the prediction of a new testing sample is done directly without need to be involved in the process from the beginning.

The amino acids in this thesis are represented by different sets of natural amino acids' physicochemical properties or features derived from the natural PCPs using the feature extraction technique, in order to know the effect of these features on our classification method on different sets of protein data.

In this thesis, we represent the protein sequences using different encoding methods based on the physicochemical properties of the amino acids, these encoding methods are used to examine the performance of the proposed approach.

We have tested the performance of the proposed approach on various types of protein biological features so as to ensure general applicability of this approach regardless of the protein problem uniqueness.

Chapter 4

Data and Methods

This chapter covers the methodology used in this thesis that aims at enhancing the performance of the prediction process, when dealing with protein sequences. We start with introducing a description of the benchmark data and a general description of our proposed approach. We define sets of descriptors that will be used in our approach and study different encoding methods. After that we clarify how to predicate new samples using our approach and demonstrates the method for measuring and evaluating performance of the classification.

4.1 Datasets

Three datasets of proteins are used to examine the performance of the proposed approach on various types of protein biological features so as to ensure the general applicability of this approach, these dataset are: Membrane proteins dataset is used as a benchmark dataset for full protein sequences, Caspase and MHC class II datasets are used as benchmark datasets for peptide sequences. These benchmarks are explained in the next subsections.

4.1. DATASETS

4.1.1 Membrane proteins benchmark

Membrane proteins are the most important proteins, it helps the cells to communicate with the surroundings, they determine whether the immune system recognizes the cell as foreign or not, and they play an important role in monitoring the processes of life [2]. Membrane proteins are embedded on one side of the cell membrane, either on the outer surface or the interior wall, the discrimination of the outer proteins from the inner is of medical importance as well as genome sequencing necessity [2].

The non-redundant dataset constructed by Park and coworkers [46] are used to study the performance of our proposed approach for the full protein sequences (different length sequences), it contains 208 outer membrane proteins (OMPs), 673 globular proteins, and 206 α -helical membrane proteins [46].

In our study, we emphasis on identifying the OMPs from inner membrane proteins, so OMPs and the α -helical membrane proteins are selected from the Parks dataset to construct a benchmark contains two classes, where the OMPs represent the positive class and the α -helical membrane proteins represent the negative class. Figure 4.1 represents an example of OMPs and α -helical membrane protein sequences.

4.1.2 MHC class II benchmark

The Major Histocompatibility Complex (MHC) is a large genomic region or gene family found in most vertebrates that encodes MHC molecules [58], and it plays an critical role in the immune system and autoimmunity [58]. Only a small fraction of the possible peptides that can be generated from proteins actually generates an immune response [58]. MHC molecules act as receptors for peptides derived from foreign antigens as well as self peptides

4.1. DATASETS

	Protein sequence	Label
OMP	MGLFLRLVLSLVLSLFSSTKANTIFTFKQALANAYRNNPELQAEIDKAQAMRGAFIQSG LYPNPQLNLTAEENFGGSGVYSSYESAETTASVTQPIPLGHRLOYLQKATYADYLTSLASI KVQKTVLYMAVGNAYVDALYAEQWHKVTKKLTKLNQDQIVVAIERRVKAGAGAEIDL RLAQVRLGEARIQETKASRDALLQRARLARLLGYGLRIDKPLVDKGLPGLTLDWSELIK KLPOSPQLVQMQLQLQARRATITAVKKSVPDLNQLGGRHFSDDGNSAAMVSAFAEV PYNRRNQGKINAEAQYTQAAHEFQSTRLEVRQNVYAVFLQAQSQSYEANLVTDSLLPS ARKSILQAQEGYQMGRTYVELYALSTLYEEEPHYQQAADYHKSILQMTGLLGLLEP IKESQ	1
	MIDTQYSLAATQAAIPSEPIAPGAAGRSVGTFQAAADLPQVPAARAADRVELNAPRQVLD PVRMEAGSELDSSVELLILFRIAQKARELGVLQRDNENQSIHQAQKQVDEMRSGATL MIANLAVIAGV GALASAVVSGL GALKNGKAI SQE KTLQKNIDGNELIDAKMIALGKTS D EDRKIVGKVAADQVQDSVALRAAGRAFESRNGALQVANTVQS FVQMANASVQVRQ GESQASAREGEVNATIGQSQKQKVEDQMSFDAGFMKDV LQLIQOYTQSHNQAWRAA GVV	1
α -helical membrane proteins	MDLATLLGLIGGFVYIMAMV LGGSIGMIFVDVTSILIVVGGSI FVYLMKFTMGQFFGATK IAGKAFMFKADEPEDLIAKIVEMADAARKGGFLALEEMEINNTFMQKGDILLVDGHDAD VYRAALKKDIALTDERHTQGTGVTRAFGDVAPAMGMIGTLVGLVAMLSMDDPKAIGF AMAVALLTILYGAILSNMIVFFPIADKLSLRDQETLNRRLIMDGVLAIQDQGNPRVDSY LKNYLNKGRAL EIDE	-1
	MLYGFSGVILQGANVLELALSSVVLAVLIGLVGAGAKLSQNRV TOLIFE GYTTILIRGVP DLVLMILLIFYGLQIALNVVTDLSLGDIDIDPMVAGHITLGFY GAYFTEFRGAFMAVPK GHIEAATAFGFTHQQTFRRIMFPAMRYALP GIGNNWQVILKATALV SLLGLEDDVYKA TQLAGKSTWEPFYFAVCGLIYLVFTTVSNGVLLILERRY SVGVRADL	-1

Figure 4.1: Example of membrane protein sequences. The first two sequences represent the OMP that have a positive label and the next two sequences represent the α -helical membrane proteins that have a negative label, where all sequences have the same different lengths.

and enable the long-term display of antigens on the cell surface [50]. There are two major types of MHC molecules are involved in the peptide binding process; class I MHC and class II MHC [50]. Prediction of peptide-MHC binding represents an important goal in bioinformatics, because of their role in the immune system. Prediction of peptides binding to an MHC class II molecule is more difficult than MHC class I due to different length of the binding peptides is longer than 9mer (sequences contain more than 9 amino acids) [50].

Peptide datasets used in this study are available from the NetMHCII 2.2 server [27]. The dataset was used in this study is DRB1*0101 dataset which contains 5166 peptides.

When classifying the peptides into binders and non-binders, a threshold value is used. This means that peptides with binding affinity values greater than 0.426 are classified as binders [27]. The main characteristics of this dataset, that it contains 5166 sequences, 1656 are non binders, and 3510 are binders. The second characteristic is that the longest sequence contains

4.1. DATASETS

37 amino acid. Figure 4.2 represents an example of binder and non-binder sequences of MHC-II.

This dataset is used in this thesis to study the performance of our approach on peptide sequences that have different lengths.

	MHC II peptide	Label
Binder sequences	VDSYYSLLMPILTLT	1
	VPIVDSYYSLLMPIL	1
	YYSLLMPILTLTRAL	1
Non-binder sequences	QARQFDQQVWEKYGH	-1
	TVVEFDSIPNKEHIP	-1
	SLLMPILTLTRALAA	-1

Figure 4.2: Example of MHC-II peptides. The first two sequences represent the binder sequences that have a positive label and the next two sequences represent the non-binder sequences that have a negative label, where these sequences vary in their lengths.

4.1.3 Caspase-3 benchmark

Caspases are a family of cysteiny proteases that regulate apoptosis (cell death) and other biological processes. Caspase-3 is considered the central executioner member of this family with a wide range of substrates [7]. It has a major role in programmed cell death as well as other vital cellular processes. As a specified-opeptidase, caspase3 cleaves its substrates after aspartic acid residue 'D'. Although the presence of the amino acid D in the target sequence is a mandatory condition yet it is not enough for recognition and cleavage by this caspase [7]. Identification of Caspase3 novel substrates is crucial to advance the understanding of the biological roles of this important enzyme.

In our study, a dataset of Caspase3 human substrates is used [7], this dataset contains 247 mapped cleavage sites and these sequences represent a positive data. While the negative data are 247 non-cleaved peptides extracted randomly and contained aspartic acid residue 'D' but outside the Caspase3

4.2. GENERAL DESCRIPTION OF THE PROPOSED APPROACH

cleaved site.

The main characteristics of Caspase sequences, that all sequences have the same lengths from 14 amino acids. Figure 4.3 represents an example of cleaved and non-cleaved peptides of Caspase-3.

This dataset is used in this thesis to study the performance of our approach on peptide sequences that have same lengths.

	Cpase peptide	Label
Cleaved peptides	VRLLQDSVDFSLAD	1
	VSDPEDITDCPRTP	1
	WESPLDEVDKMCHL	1
Non-cleaved peptides	IEKGASDEDIKKAY	-1
	AALLTDIEDMLQLI	-1
	CECNKILDVNDNI	-1

Figure 4.3: Example of Caspase-3 peptides. The first two sequences represent the cleaved peptides that have a positive label and the next two sequences represent the non-cleaved peptides that have a negative label, where all sequences have the same length.

4.2 General description of the proposed approach

Our approach for enhancing the classification performance is depending on clustering the sequences into groups then using the classification algorithm for each cluster. The protein sequences are converted from string sequences into numeric sequences by different encoding methods using descriptors based on the physicochemical properties of amino acids.

Formally, given a protein dataset $S = s_1, s_2, \dots, s_n$, that we wish to classify through mapping them into a set of biological labels y_1, y_2, \dots, y_n . We first encode S by applying an encoding transform $T = E(s_i, p) \forall i = 1, 2, \dots, n$. The new set T is now of size $p \times n$. Next we divide T into a training

4.2. GENERAL DESCRIPTION OF THE PROPOSED APPROACH

set (Tr) and testing set (Te). We apply clustering to the elements in Tr and group them into the clusters c_1, c_2, \dots, c_k based on their similarities. Next, we perform a learning process on each group of samples that belong to each cluster by a unique classifier. This means, the set of classifiers $\tau_1, \tau_2, \dots, \tau_k$ will obtain knowledge $\gamma_1, \gamma_2, \dots, \gamma_k$.

To verify the results, the following are performed: we cluster the elements of the test set Te using a distance criteria Δ . After that, we classify each element Te_i through the knowledge γ_j if $Te_i \in c_j$. We compare the result of the classifier for the sample Te_i with y_i to verify the success of that classifier.

Figure 4.4 shows a general block diagram of the proposed approach. The next sections explain the proposed approach in detail.

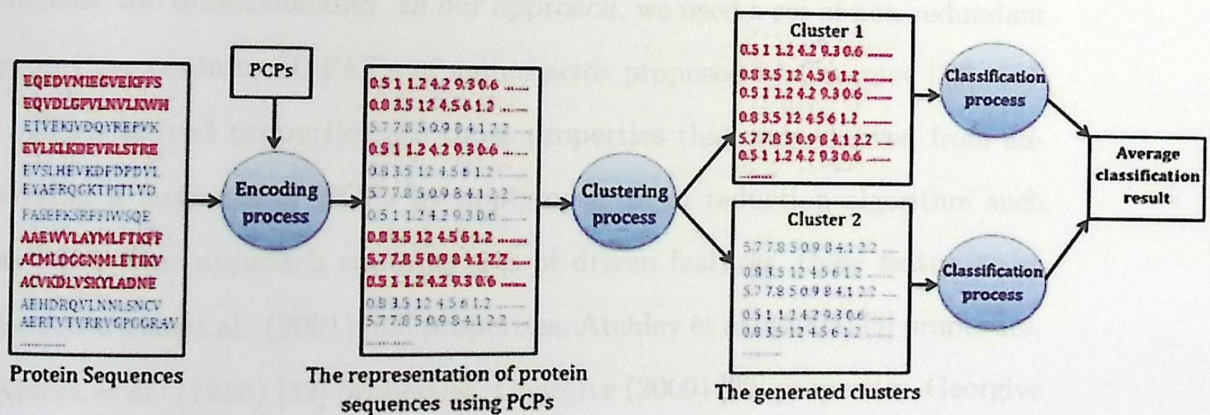


Figure 4.4: Block diagram of the proposed approach. each protein sequence encoded to numerical values using PCPs, and then the clustering algorithm used to cluster the sequences into groups, after that, each cluster classified using the classification algorithm to find the average result of classification.

4.3 Native and derived descriptors used in this study

The first step in our approach is the selection of the suitable descriptors in order to represent the amino acids of the proteins. In this thesis two sets of descriptors; Native PCPs and derived properties are used to examine the performance of the proposed approach.

The Native properties are the PCPs that represent a given measured property such as size, hydrophobicity, polarity or inferred propensity such as relative frequency to occur in an alpha helix or beta sheet. As mentioned before, the database of amino acid indices contains redundant properties that increase the dimensionality. In our approach, we used a set of non-redundant properties contains 50 PCPs of amino acids proposed by Georgiev [22].

The derived properties are those properties that were derived from analyzing a large set of PCPs by applying a given reduction algorithm such as PCA. Our approach contains sets of driven features, these features are: Venkatarajan et al. (2001) [59] properties, Atchley et al. (2005) [5] properties, Kidera et al. (1985) [32] properties, Georgiev (2009) [22] properties, Georgiev (2009) [22] BLOSUM properties, Maetschke et al. (2005) [36] properties.

These descriptors are selected based on Georgiev study [22].

4.3.1 The native properties

These properties were selected from AAindex, in which the duplicated or closely related properties were removed by an iterative procedure until 50 properties with no more than 50% correlation between them, based on Georgiev study [22]. The names of these properties are illustrated in Table 4.1, the values of these properties can be extracted from AAindex.

4.3. NATIVE AND DERIVED DESCRIPTORS USED IN THIS STUDY

Table 4.1: The description of the native properties [22]

Description of the properties
BUNA790102 alpha-CH chemical shifts (Bundi-Wuthrich, 1979)
BUNA790103 Spin-spin coupling constants 3JH _{alpha} -NH (Bundi-Wuthrich, 1979)
CHAM830102 Residuals from the best correlation of the Chou-Fasman parameter of b-sheet
The number of atoms in the side chain labelled 11 (Charton-Charton, 1983)
The number of atoms in the side chain labelled 21 (Charton-Charton, 1983)
Frequency of the 4th residue in turn (Chou-Fasman, 1978b)
Helix termination parameter at position j-2,j-1,j (Finkelstein et al., 1991)
Normalized relative frequency of double bend (Isogai et al., 1980)
pK (-COOH) (Jones, 1975)
Relative mutability (Jones et al., 1992)
Flexibility parameter for two rigid neighbors (Karplus-Schulz, 1985)
The Kerr-constant increments (Khanarian-Moore, 1980)
Normalized frequency of zeta R (Maxfield-Scheraga, 1976)
Short and medium range non-bonded energy per residue (Oobatake-Ooi, 1977)
Optimized transfer energy parameter (Oobatake et al., 1985)
Normalized frequency of alpha-helix in all-alpha class (Palau et al., 1981)
Intercept in regression analysis (Prabhakaran-Ponnuswamy, 1982)
Slope in regression analysis x 1.0E1 (Prabhakaran-Ponnuswamy, 1982)
Weights for alpha-helix at the window position of -6 (Qian-Sejnowski, 1988)
Weights for alpha-helix at the window position of -5 (Qian-Sejnowski, 1988)
Weights for beta-sheet at the window position of -3 (Qian-Sejnowski, 1988)
Weights for coil at the window position of -5 (Qian-Sejnowski, 1988)
Weights for coil at the window position of -4 (Qian-Sejnowski, 1988)
Weights for coil at the window position of 5 (Qian-Sejnowski, 1988)
Average relative fractional occurrence in AL(i) (Rackovsky-Scheraga, 1982)
Average relative fractional occurrence in ER(i) (Rackovsky-Scheraga, 1982)
Average relative fractional occurrence in A0(i-1) (Rackovsky-Scheraga, 1982)
Relative preference value at N (Richardson-Richardson, 1988)
Relative preference value at N2 (Richardson-Richardson, 1988)
Relative preference value at N3 (Richardson-Richardson, 1988)
Relative preference value at C1 (Richardson-Richardson, 1988)
Relative preference value at C (Richardson-Richardson, 1988)
Relative preference value at C (Richardson-Richardson, 1988)
Information measure for extended without H-bond (Robson-Suzuki, 1976)
Normalized frequency of isolated helix (Tanaka-Scheraga, 1977)
Normalized frequency of left-handed helix (Tanaka-Scheraga, 1977)
Normalized frequency of zeta R (Tanaka-Scheraga, 1977)
Relative population of conformational state A (Vasquez et al., 1983)
Electron-ion interaction potential (Veljkovic et al., 1985)
Free energy change of alpha(Ri) to alpha(Rh) (Wertz-Scheraga, 1978)
Normalized positional residue frequency at helix termini Nc (Aurora-Rose, 1998)
Normalized positional residue frequency at helix termini C@ (Aurora-Rose, 1998)
Normalized positional residue frequency at helix termini C40 (Aurora-Rose, 1998)
Amphiphilicity index (Mitaku et al., 2002)
Electron-ion interaction potential values (Cosic, 1994)
Hydrophobicity coefficient in RP-HPLC, C8 with 0.1%TFA/MeCN/H2O(Wilce et al., 1995)
Hydrophobicity coefficient in RP-HPLC, C4 with 0.1%TFA/MeCN/H2O(Wilce et al., 1995)
Hydrophobicity coefficient in RP-HPLC, C18 with 0.1%TFA/2-PrOH/MeCN/H2O(Wilce et al., 1995)
Hydrophobicity coefficient in RP-HPLC, C18 with 0.1%TFA/2-PrOH/MeCN/H2O(Wilce et al., 1995)
Linker propensity from 2-linker dataset (George-Heringa, 2003)
Linker propensity from long dataset (linker length is greater than 14 residues)(George-Heringa, 2003)

4.3. NATIVE AND DERIVED DESCRIPTORS USED IN THIS STUDY

Table 4.2: Values of Kidera's factors [32]

AA	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
A	-1.56	-1.67	-0.97	-0.27	-0.93	-0.78	-0.20	-0.08	0.21	-0.48
D	0.58	-0.22	-1.58	0.81	-0.92	0.15	-1.52	0.47	0.76	0.70
C	0.12	-0.89	0.45	-1.05	-0.71	2.41	1.52	-0.69	1.13	1.10
E	-1.45	0.19	-1.61	1.17	-1.31	0.40	0.04	0.38	-0.35	-0.12
F	-0.21	0.98	-0.36	-1.43	0.22	-0.81	0.67	1.10	1.71	-0.44
G	1.46	-1.96	-0.23	-0.16	0.10	-0.11	1.32	2.36	-1.66	0.46
H	-0.41	0.52	-0.28	0.28	1.61	1.01	-1.85	0.47	1.13	1.63
I	-0.73	-0.16	1.79	-0.77	-0.54	0.03	-0.83	0.51	0.66	-1.78
K	-0.34	0.82	-0.23	1.70	1.54	-1.62	1.15	-0.08	-0.48	0.60
L	-1.04	0.00	-0.24	-1.10	-0.55	-2.05	0.96	-0.76	0.45	0.93
M	-1.40	0.18	-0.42	-0.73	2.00	1.52	0.26	0.11	-1.27	0.27
N	1.14	-0.07	-0.12	0.81	0.18	0.37	-0.09	1.23	1.10	-1.73
P	2.06	-0.33	-1.15	-0.75	0.88	-0.45	0.30	-2.30	0.74	-0.28
Q	-0.47	0.24	0.07	1.10	1.10	0.59	0.84	-0.71	-0.03	-2.33
R	0.22	1.27	1.37	1.87	1.70	0.46	0.92	0.39	0.23	0.93
S	0.81	-1.08	0.16	0.42	-0.21	-0.43	-1.89	-1.15	-0.97	-0.23
T	0.26	-0.70	1.21	0.63	-0.10	0.21	0.24	-1.15	-0.56	0.19
V	-0.74	-0.71	2.04	-0.40	0.50	-0.81	-1.07	0.06	-0.46	0.65
W	0.30	2.10	0.72	1.57	1.16	0.57	0.48	0.40	2.30	0.60
Y	1.38	1.48	0.80	-0.56	0.00	-0.68	-0.31	1.03	-0.05	0.53

4.3.2 Kidera's properties

Kidera performed Factor Analysis (FA) on all available sets of physical properties of the 20 amino acids. They demonstrated that all of these data can be represented by a set of 10 property factors, these factors correlated with α -helical propensity, bulk, β -sheet propensity, and hydrophobicity [32].

The first four factors are essentially pure physical properties (Helix/bend preference, Side-chain size, Extended structure preference, and the Hydrophobicity); the remaining six factors are extracted of several physical properties [32]. Table 4.2 illustrates the values of these 10 factors.

4.3. NATIVE AND DERIVED DESCRIPTORS USED IN THIS STUDY

Table 4.3: Values of Atchley's factors [5]

AA	F1	F2	F3	F4	F5
A	-0.591	-1.302	-0.733	1.570	-0.146
C	-1.343	0.465	-0.862	-1.020	-0.255
D	1.050	0.302	-3.656	-0.259	-3.242
E	1.357	-1.453	1.477	0.113	-0.837
F	-1.006	-0.590	1.891	-0.397	0.412
G	-0.384	1.652	1.330	1.045	2.064
H	0.336	-0.417	-1.673	-1.474	-0.078
I	-1.239	-0.547	2.131	0.393	0.816
K	1.831	-0.561	0.533	-0.277	1.648
L	-1.019	-0.987	-1.505	1.266	-0.912
M	-0.663	-1.524	2.219	-1.005	1.212
N	0.945	0.828	1.299	-0.169	0.933
P	0.189	2.081	-1.628	0.421	-1.392
Q	0.931	-0.179	-3.005	-0.503	-1.853
R	1.538	-0.055	1.502	0.440	2.897
S	-0.228	1.399	-4.760	0.670	-2.647
T	-0.032	0.326	2.213	0.908	1.313
V	-1.337	-0.279	-0.544	1.242	-1.262
W	-0.595	0.009	0.672	-2.128	-0.184
Y	0.260	0.830	3.097	-0.838	1.512

4.3.3 Atchley's properties

Atchley et al. developed an approach for general use by applying FA followed by promax rotation [5], in order to compute five factors from 54 selected amino acid attributes, so each amino acid can be represented by five factors [5]. Table 4.3 illustrates the values of these 5 factors.

4.3.4 Venkatarajan's properties

Venkatarajan, et al. derived new 5 quantitative descriptors for the 20 naturally occurring amino acids using MDS of 237 physicalchemical properties [59]. Properties that correlate well with the five major components were hydrophobicity, size, preferences for amino acids to occur in α -helices, number of degenerate triplet codons and the frequency of occurrence of amino

4.3. NATIVE AND DERIVED DESCRIPTORS USED IN THIS STUDY

Table 4.4: Values of Venkatarajan's factors [59]

AA	F1	F2	F3	F4	F5
A	0.008	0.134	-0.475	-0.039	0.181
R	0.171	-0.361	0.107	-0.258	-0.364
N	0.255	0.038	0.117	0.118	-0.055
D	0.303	-0.057	-0.014	0.225	0.156
C	-0.132	0.174	0.070	0.565	-0.374
Q	0.149	-0.184	-0.030	0.035	-0.112
E	0.221	-0.280	-0.315	0.157	0.303
G	0.218	0.562	-0.024	0.018	0.106
H	0.023	-0.177	0.041	0.280	-0.021
I	-0.353	0.071	-0.088	-0.195	-0.107
L	-0.267	0.018	-0.265	-0.274	0.206
K	0.243	-0.339	-0.044	-0.325	-0.027
M	-0.239	-0.141	-0.155	0.321	0.077
F	-0.329	-0.023	0.072	-0.002	0.208
P	0.173	0.286	0.407	-0.215	0.384
S	0.199	0.238	-0.015	-0.068	-0.196
T	0.068	0.147	-0.015	-0.132	-0.274
W	-0.296	-0.186	0.389	0.083	0.297
Y	-0.141	-0.057	0.425	-0.096	-0.091
V	-0.274	0.136	-0.187	-0.196	-0.299

acid residues in β -strands [59]. Table 4.4 illustrates the values of these 5 factors. Table 4.4 illustrates the values of these 5 factors.

4.3.5 Maetschke's properties

Maetschke et al. derived a 5 factor from BLOSUM 62 to encode the peptide sequence in order to improve the single peptide cleavage site prediction [36], these 5 factors are illustrated in Table 4.5.

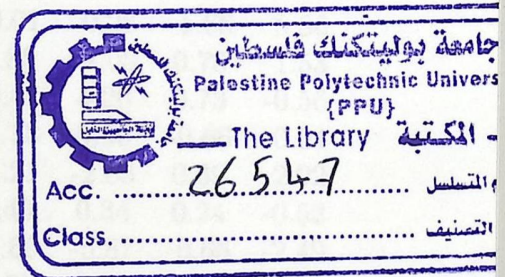
4.3.6 Georgieve's properties

Georgieve derived a 19 descriptors from 509 amino acid indices using the vari-max criterion rather than the PCA to increase the ease of interpretation [22], these 19 factors are illustrated in Table 4.6.

4.4. ENCODING PROTEIN SEQUENCES USING PCPS

Table 4.5: Values of Maetschke's Factors [36]

AA	F1	F2	F3	F4	F5
A	-0.57	0.39	-0.96	-0.61	-0.69
R	-0.40	-0.83	-0.61	1.26	-0.28
N	-0.70	-0.63	-1.47	1.02	1.06
D	-1.62	-0.52	-0.67	1.02	1.47
C	0.07	2.04	0.65	-1.13	-0.39
Q	-0.05	-1.50	-0.67	0.49	0.21
E	-0.64	-1.59	-0.39	0.69	1.04
G	-0.90	0.87	-0.36	1.08	1.95
H	0.73	-0.67	-0.42	1.13	0.99
I	0.59	0.79	1.44	-1.90	-0.93
L	0.65	0.84	1.25	-0.99	-1.90
K	-0.64	-1.19	-0.65	0.68	-0.13
M	0.76	0.05	0.06	-0.62	-1.59
F	1.87	1.04	1.28	-0.61	-0.16
P	-1.82	-0.63	0.32	0.03	0.68
S	-0.39	-0.27	-1.51	-0.25	0.31
T	-0.04	-0.30	-0.82	-1.02	-0.04
W	1.38	1.69	1.91	1.07	-0.05
Y	1.75	0.11	0.65	0.21	-0.41
V	-0.02	0.30	0.97	-1.55	-1.16



4.3.7 Georgieve's BLOSUM62 properties

Georgive performed an approach that depends on deriving new features from substitution matrix. He derived 10 factors from BLOSUM 62 substitution matrix based on Class A GPCR subfamily problem [22]. These 10 factors are illustrated in Table 4.7.

4.4 Encoding protein sequences using PCPs

Encoding methods using PCPs or extracting features from proteins is the process of representing the protein sequences as numerical sequences using a set of PCPs, in order to facilitate using of machine learning tools.

Most protein data sets contain protein sequences or peptides vary in length, and this is considered a problem when using the machine learning

4.4. ENCODING PROTEIN SEQUENCES USING PCPS

Table 4.6: Values of the Georgieva factors [22]

AA	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
A	0.57	3.37	-3.66	2.34	-1.07	-0.40	1.23	-2.32	-2.01	1.31	-1.14
R	-2.80	0.31	2.84	0.25	0.20	-0.37	3.81	0.98	2.43	-0.99	-4.90
N	-2.02	-1.92	0.04	-0.65	1.61	2.08	0.40	-2.47	-0.07	7.02	1.32
D	-2.46	-0.66	-0.57	0.14	0.75	0.24	-5.15	-1.17	0.73	1.50	1.51
C	2.66	-1.52	-3.29	-3.77	2.96	-2.23	0.44	-3.49	2.22	-3.78	1.98
Q	-2.54	1.82	-0.82	-1.85	0.09	-0.60	0.25	2.11	-1.92	-1.67	0.70
E	-3.08	3.45	0.05	0.62	-0.49	-0.00	-5.66	-0.11	1.49	-2.26	-1.62
G	0.15	-3.49	-2.97	2.06	0.70	7.47	0.41	1.62	-0.47	-2.90	-0.98
H	-0.39	1.00	-0.63	-3.49	0.05	0.41	1.61	-0.60	3.55	1.52	-2.28
I	3.10	0.37	0.26	1.04	-0.05	-1.18	-0.21	3.45	0.86	1.98	0.89
L	2.72	1.88	1.92	5.33	0.08	0.09	0.27	-4.06	0.43	-1.20	0.67
K	-3.89	1.47	1.95	1.17	0.53	0.10	4.01	-0.01	-0.26	-1.66	5.86
M	1.89	3.88	-1.57	-3.58	-2.55	2.07	0.84	1.85	-2.05	0.78	1.53
F	3.12	0.68	2.40	-0.35	-0.88	1.62	-0.15	-0.41	4.20	0.73	-0.56
P	-0.58	-4.33	-0.02	-0.21	-8.31	-1.82	-0.12	-1.18	0.00	-0.66	0.64
S	-1.10	-2.05	-2.19	1.36	1.78	-3.36	1.39	-1.21	-2.83	0.39	-2.92
T	-0.65	-1.60	-1.39	0.63	1.35	-2.45	-0.65	3.43	0.34	0.24	-0.53
W	1.89	-0.09	4.21	-2.77	0.72	0.86	-1.07	-1.66	-5.87	-0.66	-2.49
Y	0.79	-2.62	4.11	-0.63	1.89	-0.53	-1.30	1.31	-0.56	-0.95	1.91
V	2.64	0.03	-0.67	2.34	0.64	-2.01	-0.33	3.93	-0.21	1.27	0.43

AA	F12	F13	F14	F15	F16	F17	F18	F19
A	0.19	1.66	4.39	0.18	-2.60	1.49	0.46	-4.22
R	2.09	-3.08	0.82	1.32	0.69	-2.62	-1.49	-2.57
N	-2.44	0.37	-0.89	3.13	0.79	-1.54	-1.71	-0.25
D	5.61	-3.85	1.28	-1.98	0.05	0.90	1.38	-0.03
C	-0.43	-1.03	0.93	1.43	1.45	-1.15	-1.64	-1.05
Q	-0.27	-0.99	-1.56	6.22	-0.18	2.72	4.35	0.92
E	-3.97	2.30	-0.06	-0.35	1.51	-2.29	-1.47	0.15
G	-0.62	-0.11	0.15	-0.53	0.35	0.30	0.32	0.05
H	-3.12	-1.45	-0.77	-4.18	-2.91	3.37	1.87	2.17
I	-1.67	-1.02	-1.21	-1.78	5.71	1.54	2.11	-4.18
L	-0.29	-2.47	-4.79	0.80	-1.43	0.63	-0.24	1.01
K	-0.06	1.38	1.78	-2.71	1.62	0.96	-1.09	1.36
M	2.44	-0.26	-3.09	-1.39	-1.02	-4.32	-1.34	0.09
F	3.54	5.25	1.73	2.14	1.10	0.68	1.46	2.33
P	-0.92	-0.37	0.17	0.36	0.08	0.16	-0.34	0.04
S	1.27	2.86	-1.88	-2.42	1.75	-2.77	3.36	2.67
T	1.91	2.66	-3.07	0.20	-2.20	3.73	-5.46	-0.73
W	-0.30	-0.50	1.64	-0.72	1.75	2.73	-2.20	0.90
Y	-1.26	1.57	0.20	-0.76	-5.19	-2.56	2.87	-3.43
V	-1.71	-2.93	4.22	1.06	-1.31	-1.97	-1.21	4.77

4.4. ENCODING PROTEIN SEQUENCES USING PCPS

Table 4.7: Values of Georgieve BLOSUM62 factors [22]

AA	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
A	0.077	-0.916	0.526	0.004	0.240	0.190	0.656	-0.047	1.357	0.333
R	1.014	0.189	-0.860	-0.609	1.277	0.195	0.661	0.175	-0.219	-0.520
N	1.511	0.215	-0.046	1.009	0.120	0.834	-0.033	-0.570	-1.200	-0.139
D	1.551	0.005	0.323	0.493	-0.991	0.010	-1.615	0.526	-0.150	-0.282
C	-1.084	-1.112	1.562	0.814	1.828	-1.048	-0.742	0.379	-0.121	-0.102
Q	1.094	0.296	-0.871	-0.718	0.500	-0.080	-0.442	0.202	0.384	0.667
E	1.477	0.229	-0.670	-0.355	-0.284	-0.075	-1.014	0.363	0.769	0.298
G	0.849	0.174	1.726	0.093	-0.548	1.186	1.213	0.874	0.009	0.242
H	0.716	1.548	-0.802	1.547	0.350	-0.785	0.655	-0.076	-0.186	0.990
I	-1.462	-1.126	-0.761	0.382	-0.599	0.276	-0.132	0.198	-0.216	0.207
L	-1.406	-0.856	-0.879	-0.172	0.032	0.344	0.109	0.146	-0.436	-0.021
K	1.135	-0.039	-0.802	-0.849	0.819	0.097	0.213	0.129	0.176	-0.850
M	-0.963	-0.585	-0.972	-0.528	0.236	0.365	0.062	0.208	-0.560	0.361
F	-1.619	1.007	-0.311	0.623	-0.549	0.290	-0.021	0.098	0.433	-1.288
P	0.883	-0.675	0.382	-0.869	-1.243	-2.023	0.845	-0.352	-0.421	-0.298
S	0.844	-0.448	0.423	0.317	0.200	0.541	0.009	-0.797	0.624	-0.129
T	0.188	-0.733	0.178	-0.012	0.022	0.378	-0.304	-1.958	0.149	0.063
W	-1.577	2.281	1.166	-1.610	0.122	0.239	-0.542	-0.398	-0.349	0.499
Y	-1.142	1.740	-0.582	0.747	-0.119	-0.475	0.241	-0.251	0.713	-0.251
V	-1.127	-1.227	-0.633	0.064	-0.596	0.158	0.014	0.016	0.251	0.607

tools, several methods used to overcome this problem; such as: remove some amino acids from sequences to unify the lengths of the sequences, another method depend on adding unused character (e.g. : J or B) to complete the sequences in order to unify the lengths.

In this thesis we used the encoding methods to represent the protein sequences numerically and to unify the length of the sequences, so we selected encoding methods that consider effective for representation and can be used to unify the lengths of the sequences. These methods are:

- PseAAC encoding: probably the most used encoding for proteins [42], it represents a protein sequence with a discrete model without completely losing its sequence order information. PseAAC is chosen in this study because it formed from weighted sums of amino acid composi-

4.5. CLASSIFICATION BASED ON CLUSTERING

tions, physicochemical square correlations and combination of amino acid compositions and dipeptide composition [42]. Therefore the feature vector is composed of 20 (from AC) + λ (correlation factors, see Section 2.2).

- CTD encoding: is the famous encoding of proteins that depending on distributing amino acids into groups based on their PCPs [44]. The feature vector is composed of 21 (from Composition) + 21 (from transition) + 105 (from distribution) for all sequences regardless their lengths.

These encoding schemas; PseAAC and CTD are selected to help in unifying the lengths, where PseAAC considered as complicated method and CTD as a method depending on distributing amino acids into groups based on their PCPs. From simple encoding methods we choose the method that depending on representing each amino acid numerically as a set of different physicochemical properties, this method is not suitable for sequences have various lengths but it is a good choice for the dataset that have sequences with fixed lengths, so it uses only for Caspase benchmark.

These encoding methods are used to examine the performance of our proposed approach using different methods.

4.5 Classification based on clustering

In this thesis, the clustering is used before the classification in order to enhance the performance of protein attributes prediction. In this section we provide a description of the proposed approach in order to explain how clustering can be used before the classification. The proposed approach consists of the following steps:

4.5. CLASSIFICATION BASED ON CLUSTERING

- Data division step: to divide the data into k -fold cross validation.
- Clustering step: to cluster the training set into N clusters.
- Distribution step: to distribute the testing data to the generated clusters.
- Classification step: to apply the classification algorithm for all generated clusters.

These steps can be clarified as follows:

4.5.1 Data division step

Let (X, Y) be an input data set where $X = \{x_1, x_2, x_3, \dots, x_n\}$, $x_i = \{x_{1i}, x_{2i}, \dots, x_{di}\}$, d is the dimension of the properties for each amino acids, and $Y = \{y_1, y_2, y_3, \dots, y_n\}$ where $y_j \in (-1, 1)$. These data are divided into M sets using the cross validation.

The main idea of the cross validation is to split the data, once or several times, for estimating the risk of each algorithm: part of the data (the training data) is used for training each algorithm, and the remaining part (the testing data) is used for evaluation of the algorithm [4]. Figure 4.5 illustrates an example of 3-fold cross validation, the dataset is divided into three groups, two of them are used for training and the rest is used for testing the method. The same procedure is repeated for three times and the average is computed for obtaining the accuracy.

4.5.2 Clustering step

In this step the training data for each fold is grouped into C clusters using a clustering algorithm. In our approach we used the K-mean algorithm to

4.5. CLASSIFICATION BASED ON CLUSTERING

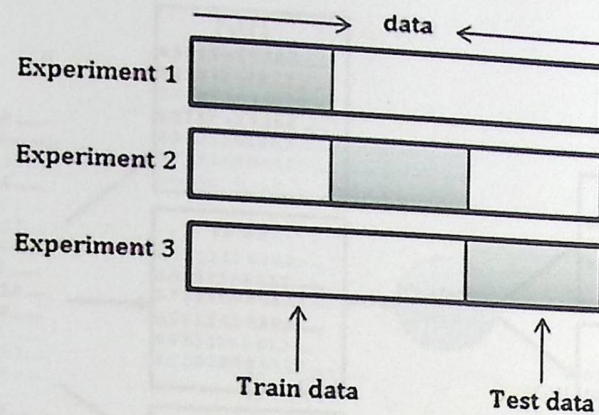


Figure 4.5: Example of cross validation. This example illustrates the 3-fold cross validation, where the data is divided into three groups, two of them are used for training and the rest is used for testing the method. The white part of the data represents the train data and the gray scale represents the test data.

cluster the data, this algorithm was chosen due to its simplicity, and based on literature [49, 33]. Figure 4.6 illustrates an example of this step, this figure shows two clusters resulted from applying the clustering algorithm for one fold of the data.

4.5.3 Distribution step

After the clustering step ends, the testing and the training data should be prepared for the classification process. The distribution step concerns on testing data. In this step the testing data distributes to the clusters as follows:

- For each cluster we have a centroid point.
- Compute the Euclidean distance between each test sample and each centroid of the generated clusters
- The testing sample relates to the cluster have the minimum distance between it and the centroids.

4.5. CLASSIFICATION BASED ON CLUSTERING

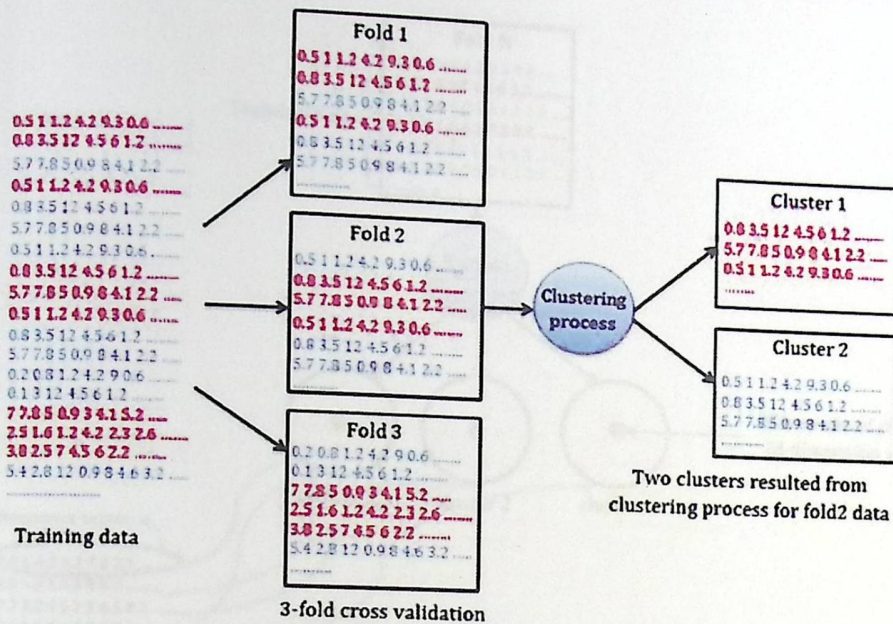


Figure 4.6: Clustering step. Two clusters generated by applying K-mean algorithm on the Fold 2, the clustering algorithm should apply on all folds

- This step is done for all folds in the approach.

Figure 4.7 illustrates the distribution step, in this figure the training data in fold N grouped into 3 clusters, each cluster has a centroid, and then the testing data distributes to the clusters based on the minimum Euclidean distance between the centroids of the clusters and the testing sample.

4.5.4 Classification step

Each cluster contains a test and train data, so we can apply a classification algorithm on each cluster. In our approach the SVM was used to classify the data, because it is one of the most powerful classification techniques that was successfully applied to many real world problems, it has proven a great success in many areas, such as protein classification and face recognition [6], and it's suitable for unbalanced data. Figure 4.8 illustrates the classification step, in this figure the classifier applied in all clusters and the result of the case is the average classification results from the clusters.

4.6. PREDICTION OF A NEW TESTING SAMPLE

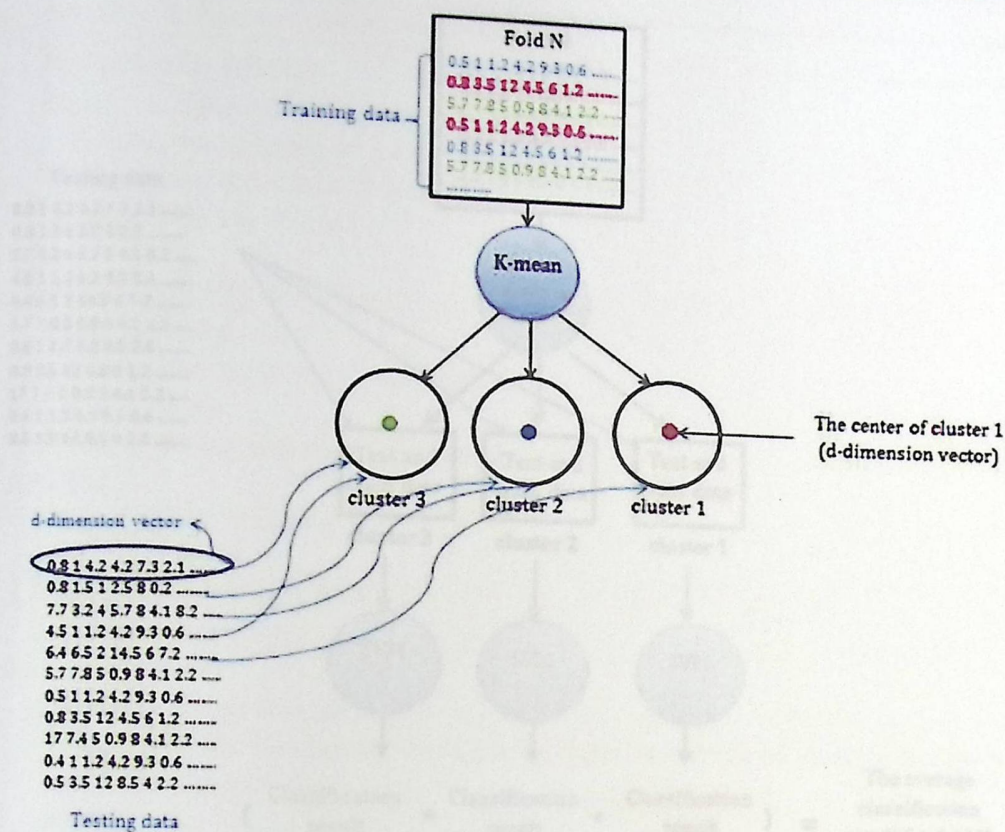


Figure 4.7: Distribution step. The distribution of the testing data into three clusters generated from the training data of Fold N, the distribution is done using the Euclidean distance

The proposed algorithm is summarised in Algorithm 2.

4.6 Prediction of a new testing sample

After the SVM trained then the prediction of new sample in our approach is done as follows:

1. The new sample (protein sequence) is encoded using the selected encoding method.
2. Finding the Euclidean distance between the sample and the centroid of each cluster.
3. Selecting the cluster that has a minimum distance with the sample.

4.7. PERFORMANCE EVALUATION

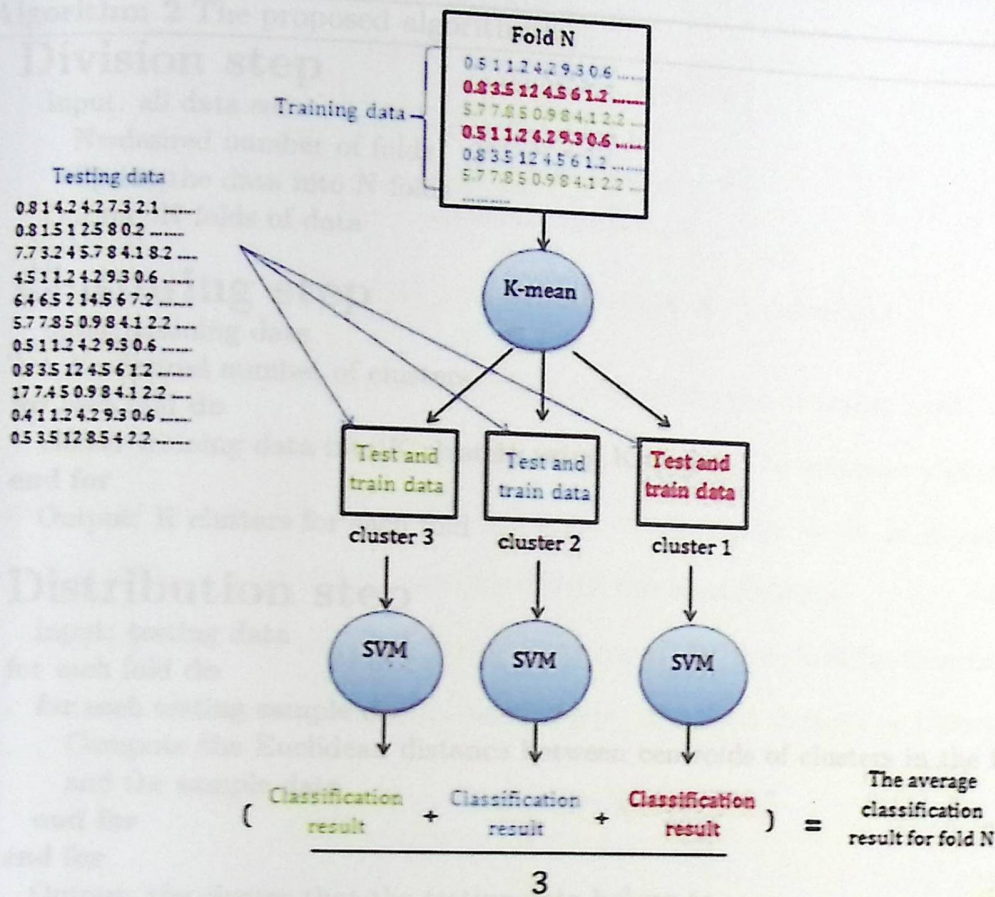


Figure 4.8: The classification step. In this step, the SVM classifier is applied on each cluster, and then an average result is generated from the results of classifier on all clusters.

- Determining the label of the new sample based on the training result of the selected cluster.

That means the prediction of a new testing sample is done directly without need to be involved in the process from the beginning.

4.7 Performance evaluation

There are different methods for measuring and evaluating performance of the classifier. Here we introduce description for the accuracy measure that is used in our experiments.

4.7. PERFORMANCE EVALUATION

Algorithm 2 The proposed algorithm

Division step

Input: all data set
N=desired number of folds
Divide the data into N folds
Output: N folds of data

Clustering step

Input: training data
K=desired number of clusters
for each fold do
 cluster training data into K clusters using K-mean
end for
Output: K clusters for each fold

Distribution step

Input: testing data
for each fold do
 for each testing sample do
 Compute the Euclidean distance between centroids of clusters in the fold
 and the sample data
 end for
end for
Output: the cluster that the testing data belong to.

Classification step

Input: training and testing data of previous step
for each cluster do
 Train SVM classifier based on the training data for N folds
 Classify the testing data belong to each cluster for N folds
end for
 Compute the average result for the classification for all clusters.
Output: predicate labels of testing data and the average value of the performance measure of classifier

All measures of performance are based on four possible outcomes obtained from applying the classifier on the test data, Figure 4.9 illustrates these values, where: *TP* means the instance that is positive in truth and classified as positive, *FN* means the instance that is positive in truth and classified as negative, *TN* means the instance that is negative in truth and classified as negative, and *FP* means the instance that is negative in truth and classified as positive [11].

4.7. PERFORMANCE EVALUATION

		Predicted	
		Positive	Negative
Truth	Positive	TP	FN
	Negative	FP	TN

Figure 4.9: The four possible outcome of the classifier

The accuracy was chosen in our work to evaluate the classifier result and it is efficient in our approach because we do not need to compare different classification algorithms, but we need it to compare the result of classifier with and without applying clustering before the classification.

Accuracy (ACC) is an evaluation measurement for the classifier that takes into account all true classification results [19], it can be defined as follows

$$ACC = \frac{TP + TN}{TP + FN + FP + TN} \quad (4.1)$$

descriptors. 5 of them are derived properties, and one set is native properties, those sets mentioned in Section 4.2. Three-fold cross validation was used for testing and comparing the results.

Two environments were used to implement this thesis: Java and Matlab. In Java we used the Java machine learning library JavaML [4], the Composition, Transition, Distribution encoding using the Ngram library [47], and we used the Matlab Statistics Toolbox for the SVM

Chapter 5

Experiments and Results

5.2 Results from full protein sequences

In this chapter we present the experimental results of applying our approach. In the first section, we show the general settings and parameter adjustments of our experiments. In the second section, we introduce the results generated from the full protein sequences using the membrane proteins benchmark. In the third section, we introduce the results generated from the peptide sequences using MHC class II (MHC-II) and Caspase benchmarks. In the fourth section, the time performance based on the proposed approach is discussed. Finally, we discuss the performance of selected properties, encoding method in the proposed approach, the performance of our approach for benchmarks, and the performance of training time.

5.1 Experimental settings

In our experiments we used the K-mean algorithm to cluster the data, the SVM algorithm to classify the data, and two main encoding methods; PseAAC and CTD to represent the protein sequences.

For PseAAC we set $w = 0.15$ for all experiments and $\lambda = 30$ [21] for full protein sequences and $\lambda = 3$ for peptide sequences, we used 7 sets of

5.2. RESULTS FROM FULL PROTEIN SEQUENCES

descriptors, 6 of them are derived properties, and one set is native properties, these sets mentioned in Section 4.2. Three-fold cross validation was used for testing and comparing the results.

Two environments were used to implement this thesis; Java and Matlab, for implementation of k-mean we used the Java machine learning library (Java-ML) [1], the Composition, Transition, Distribution encoding using the Biojava library [47], and we used the Matlab Statistics Toolbox for the SVM implementation [38].

5.2 Results from full protein sequences

The membrane proteins benchmark as full protein sequences was used in testing our approach. Two encoding methods: the PseAAC, and the CTD were used to encode the sequences in this benchmark. A description of the experiments that were done will be introduced here.

Results From PseAAC encoding method

Each protein sequence in the benchmark was encoded using the PseAAC based on Equations 2.10 to 2.13. The feature vector for each sequence was 20 (AC properties) + 30 (correlation factors from $\lambda = 30$), and we used the 7 sets of properties.

Figure 5.1 shows the accuracy results of 3 fold cross validation test for SVM classifier based on K-mean clustering using the previous 7 sets and PseAAC, the experiments were done using different numbers of clusters of training data, ranging from 2 to 10. We stopped at 10 clusters because after this number of clusters some experiments produced homogeneous clusters (clusters contain data from the same classes) that can not be classified. The

5.2. RESULTS FROM FULL PROTEIN SEQUENCES

result of one cluster of data means a result of classification without clustering of the data.

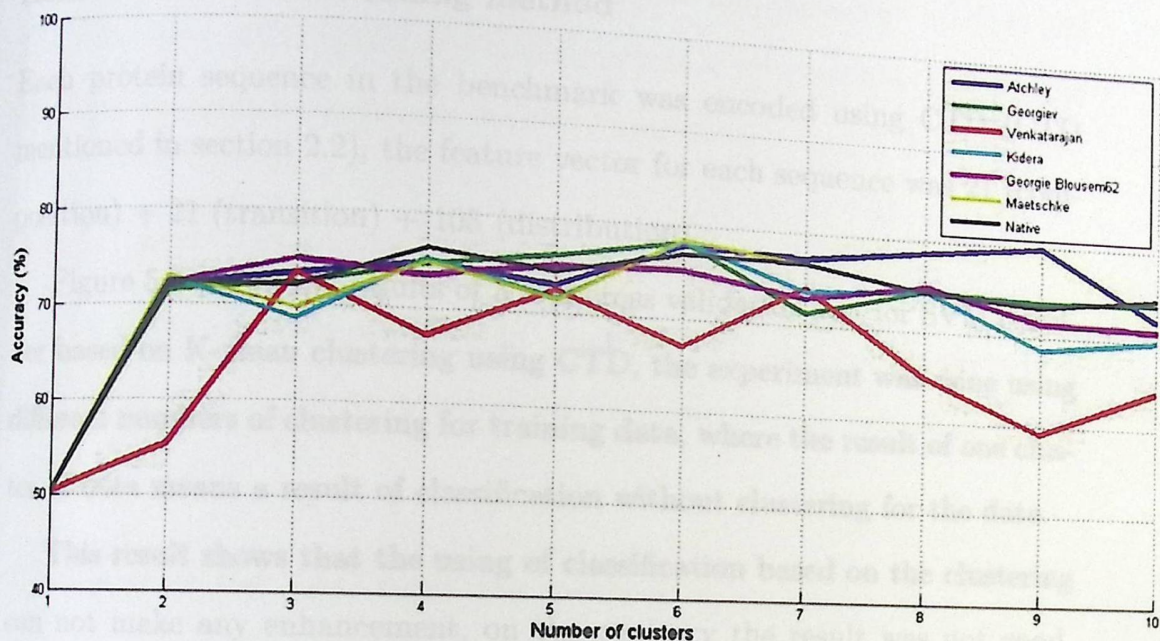


Figure 5.1: Accuracy of SVM for membrane proteins using PseACC. The training data divided into different numbers of clusters (range from 2 to 10), one cluster of training data means a classification without clustering. The x-axis represents the number of clusters and the y-axis represents the accuracy of the classification.

These results show that the using of classification without the clustering gives not good results compared with classification based on clustering. Once the data is split the accuracy risen dramatically, when the data divided into two clusters enhanced the accuracy results approximately by 20% for all sets of properties except Venkatarajan that enhanced the result by 5% only, but its still better than using classification without clustering.

All sets of properties behaved the same with different number of clusters except Venkatarajan, but the highest value was achieved by using the Atchley properties where the accuracy result arrived to approximately 80% at nine clusters.

The results also show that using derived or native properties gave almost

5.2. RESULTS FROM FULL PROTEIN SEQUENCES

the same effect using the PseACC encoding.

Results from CTD encoding method

Each protein sequence in the benchmark was encoded using CTD (CTD mentioned in section 2.2), the feature vector for each sequence was 21 (composition) + 21 (transition) + 105 (distribution).

Figure 5.2 shows the results of 3 fold cross validation test for SVM classifier based on K-mean clustering using CTD, the experiment was done using different numbers of clustering for training data, where the result of one cluster of data means a result of classification without clustering for the data.

This result shows that the using of classification based on the clustering can not make any enhancement, on the contrary the result was not good, so CTD is not a good choice to encode membrane proteins when using our proposed approach.



Figure 5.2: Accuracy of SVM for membrane proteins using CTD. The training data divided into different numbers of clusters (range from 2 to 7), one cluster of training data means a classification without clustering.

5.3 Results from peptide sequences

Two peptide benchmarks were selected to test our approach: MHC-II and Caspase, where MHC contains peptide sequences with variable lengths and the Caspase contains sequences with fixed lengths.

5.3.1 Results from MHC-II sequences

Two encoding methods: the PseAAC, and the CTD were used to encode the MHC-II sequences, a description of the experiments that were done will be introduced below.

Results from PseAAC encoding method

Each peptide sequence in the benchmark was encoded using the PseAAC. In this experiment we used $\lambda = 3$ (due to short lengths of peptide sequences), so the feature vector for each sequence was 20 (AC properties) + 3 (correlation factors). Also the 7 sets of PCPs were used in this experiment (Section 4.2).

Figure 5.3 shows the results of 3 fold cross validation test for SVM classifier based on K-mean clustering using the previous 7 sets and PseAAC, the experiments were done using different numbers of clustering of training data, range from 2 to 20, this large range was selected due to the large number of data.

The figure shows there are no significant improvement when the classification based on clustering was used, this may be due to the complexity of MHC-II, however our approach has maintained the accuracy close to the accuracy of classification without clustering.

5.3. RESULTS FROM PEPTIDE SEQUENCES

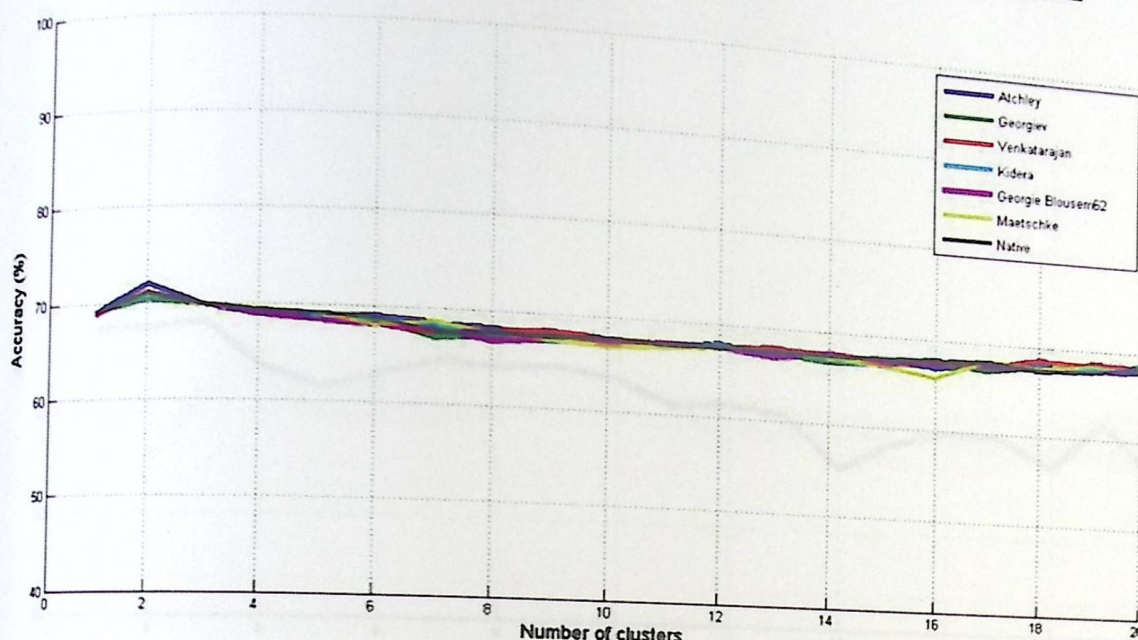


Figure 5.3: Accuracy of SVM for MHC-II sequences using PseAAC. The training data divided into different numbers of clusters (range from 2 to 20), one cluster of training data means a classification without clustering.

Results from CTD encoding method

Each sequence in the benchmark was encoded using CTD. Figure 5.4 shows the results of 3 fold cross validation test for SVM classifier based on K-mean clustering using CTD, the experiment done using different numbers of clustering of training data, where the result of one cluster of data means a result of classification without clustering for the data.

The Figure 5.4 shows that when the data divided into 2 or 3 clusters the results of accuracy increased very slightly, then when the number of clusters increased, the accuracy decreased compared to the accuracy of classification without clustering.

5.3.2 Results from Caspase sequences

Three encoding methods: PseAAC, CTD and the concatenating method were used to encode the Caspase sequences, a description of the experiments that

5.3. RESULTS FROM PEPTIDE SEQUENCES

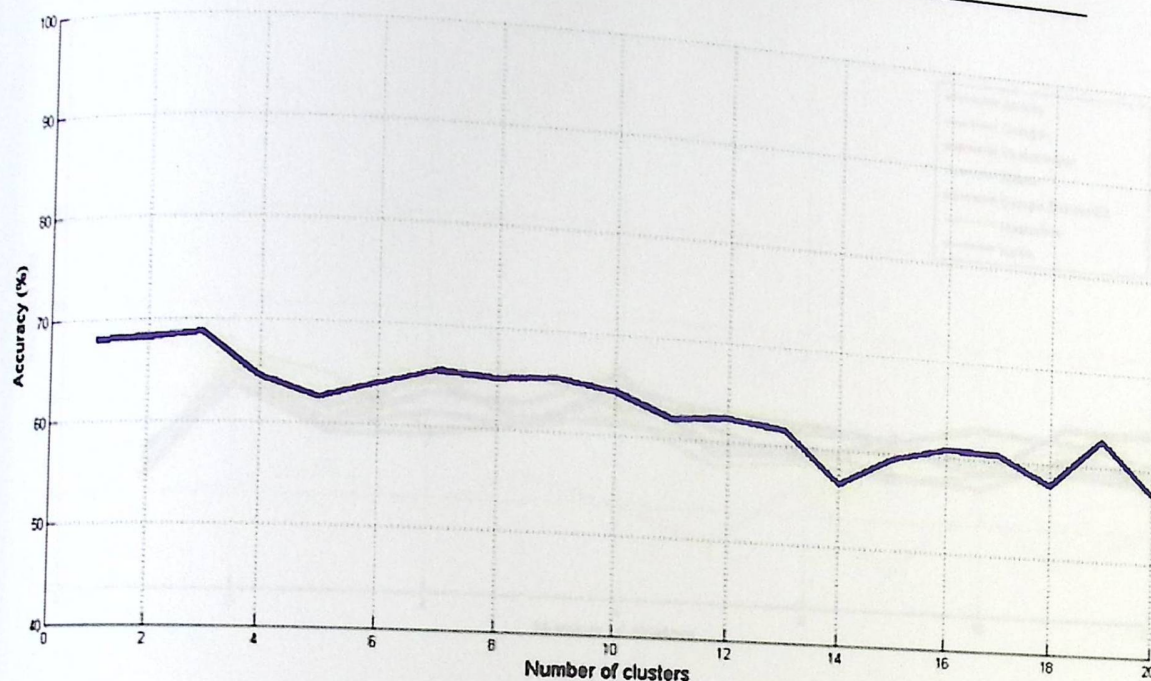


Figure 5.4: Accuracy of SVM for MHC-II sequences using CTD. The training data divided into different numbers of clusters (range from 2 to 20), one cluster of training data means a classification without clustering.

were done will be introduced below.

Results from PseAAC encoding method

Each peptide sequence in the benchmark was encoded using the PseAAC. In this experiment we used $\lambda = 3$ (due to short lengths of peptide sequences), so the feature vector was as the MHC-II feature vectors. Also the 7 sets of PCPs were used in this experiment (Section 4.2).

Figure 5.5 shows the results of 3 fold cross validation test for SVM classifier based on K-mean clustering using the previous 7 sets of properties and PseAAC, the experiments were done using different numbers of clustering of training data, range from 2 to 12. We stopped at 12 clusters because after this number of clusters some experiments produced homogeneous clusters.

From this figure we can see an improvement on the accuracy when the data divided into clusters before applying the classification, at two clusters

5.3. RESULTS FROM PEPTIDE SEQUENCES

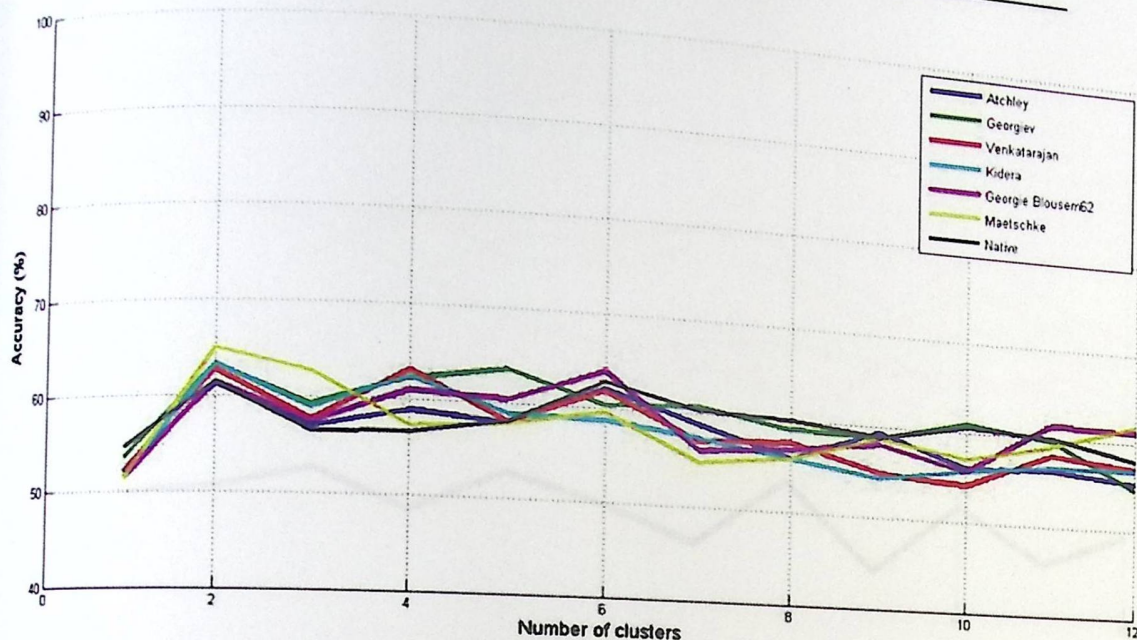


Figure 5.5: Accuracy of SVM for Caspase sequences using PseAAC. The training data divided into different numbers of clusters (range from 2 to 12), one cluster of training data means a classification without clustering.

the accuracy of the classification enhanced by 15% especially when we used Maelschkes properties. The accuracy of almost sets of properties oscillating up and down, however the classification based on clustering was the best in all cases.

Results from CTD encoding method

Each sequence in the benchmark was encoded using CTD. Figure 5.6 shows the results of 3 fold cross validation test for SVM classifier based on K-mean clustering using CTD, the experiment done using different numbers of clustering of training data, where the result of one cluster of data means a result of classification without clustering for the data.

The figure shows that the accuracy using the CTD was improved slightly ranged from (2-3)%. We can see that the CTD is not a good choice to encode the Caspase sequences for our approach.

5.3. RESULTS FROM PEPTIDE SEQUENCES

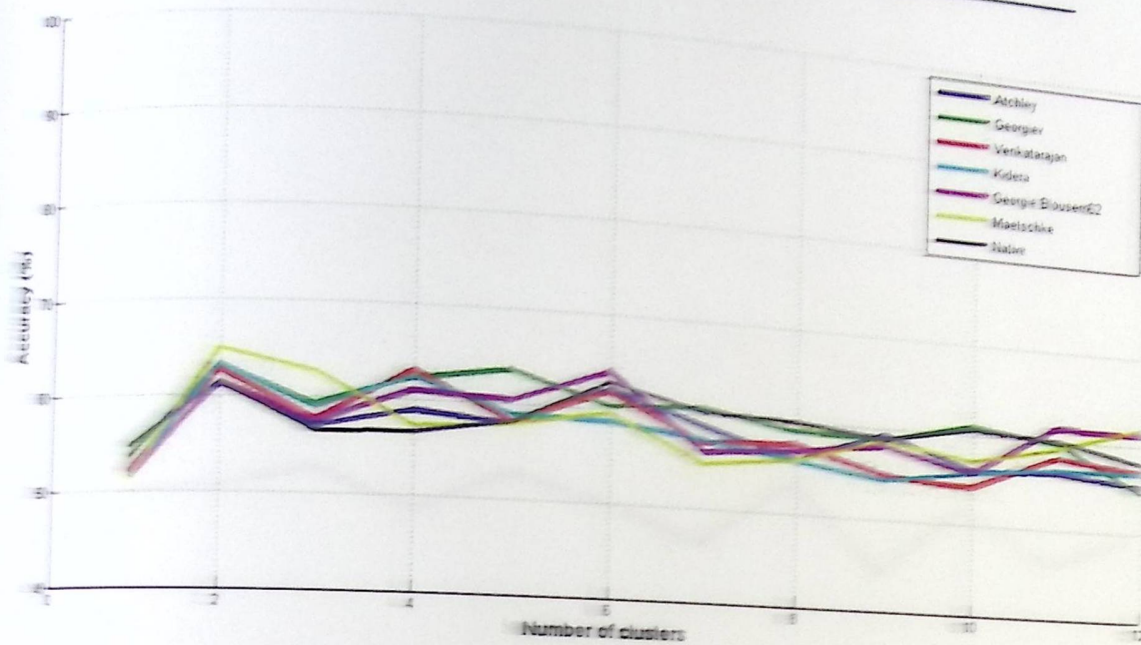


Figure 5.5: Accuracy of SVM for Caspase sequences using PseAAC. The training data divided into different numbers of clusters (range from 2 to 12), one cluster of training data means a classification without clustering.

The accuracy of the classification enhanced by 15% especially when we used Maetschkes properties. The accuracy of almost sets of properties oscillating up and down, however the classification based on clustering was the best in all cases.

Results from CTD encoding method

Each sequence in the benchmark was encoded using CTD. Figure 5.6 shows the results of 3 fold cross validation test for SVM classifier based on K-means clustering using CTD, the experiment done using different numbers of clustering of training data, where the result of one cluster of data means a result of classification without clustering for the data.

The figure shows that the accuracy using the CTD was improved slightly from 52.9%. We can see that the CTD is not a good choice to encode Caspase sequences for our approach.

5.3. RESULTS FROM PEPTIDE SEQUENCES

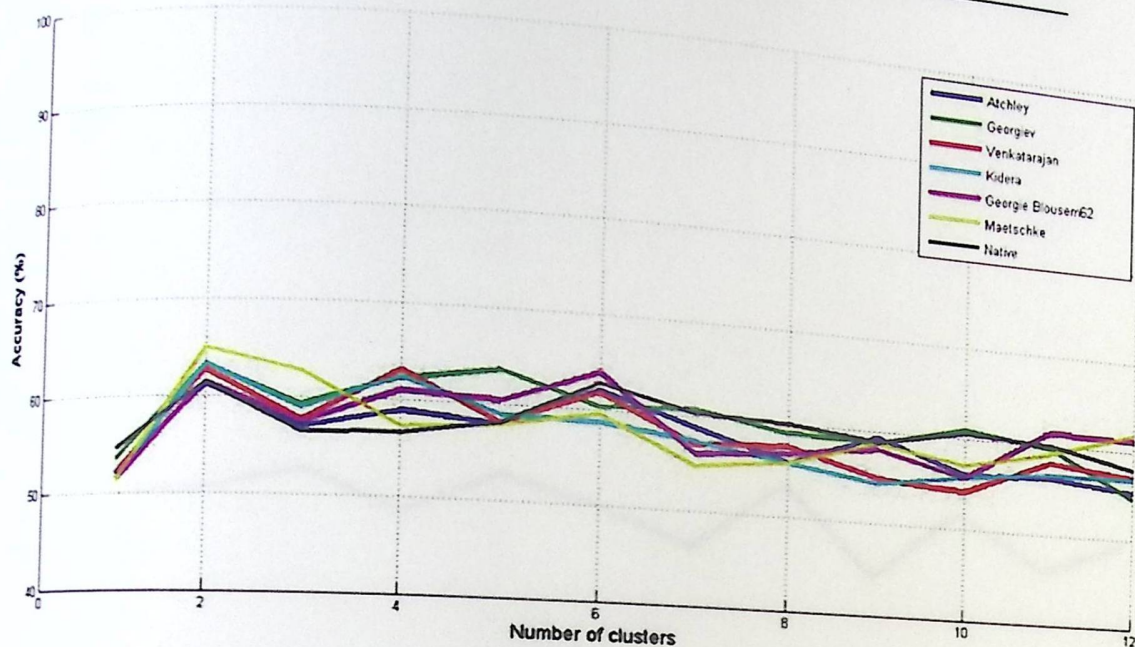


Figure 5.5: Accuracy of SVM for Caspase sequences using PseAAC. The training data divided into different numbers of clusters (range from 2 to 12), one cluster of training data means a classification without clustering.

the accuracy of the classification enhanced by 15% especially when we used Maetschkes properties. The accuracy of almost sets of properties oscillating up and down, however the classification based on clustering was the best in all cases.

Results from CTD encoding method

Each sequence in the benchmark was encoded using CTD. Figure 5.6 shows the results of 3 fold cross validation test for SVM classifier based on K-mean clustering using CTD, the experiment done using different numbers of clustering of training data, where the result of one cluster of data means a result of classification without clustering for the data.

The figure shows that the accuracy using the CTD was improved slightly ranged from (2-3)%. We can see that the CTD is not a good choice to encode the Caspase sequences for our approach.

5.3. RESULTS FROM PEPTIDE SEQUENCES

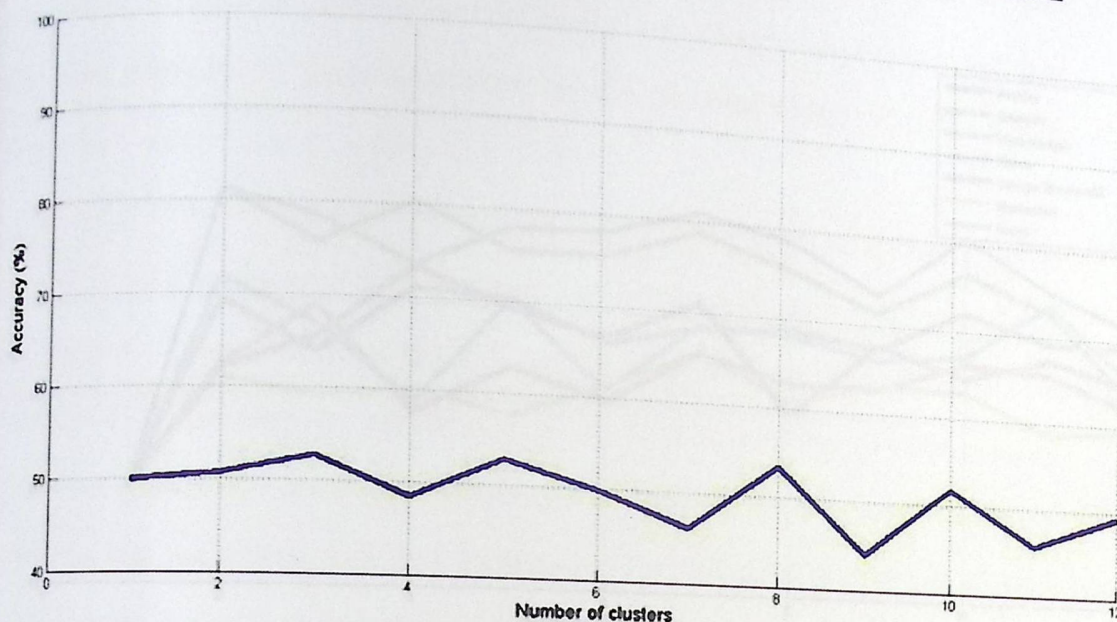


Figure 5.6: Accuracy of SVM for Caspase sequences using CTD. The training data divided into different numbers of clusters (range from 2 to 12), one cluster of training data means a classification without clustering.

Results from concatenating encoding method

This encoding method can be used when the dataset contains sequences have the same lengths, so we used it for Caspase dataset to evaluate our approach, the main feature of this method is using the original values of the properties rather than derived new values represent the original properties such as the PseAAC and CTD. Also the 7 sets of PCPs were used in this experiment.

Figure 5.7 shows the results of 3 fold cross validation test for SVM classifier based on K-mean clustering using the previous 7 sets and concatenating method, the experiments were done using different numbers of clustering of training data, range from 2 to 12.

This result shows that the using of classification based on the clustering gives significant enhancement compared with the classification without clustering when using the concatenating method. Once the data are split the accuracy risen dramatically, when the data divided into two clusters the ac-

5.4. TIME PERFORMANCE

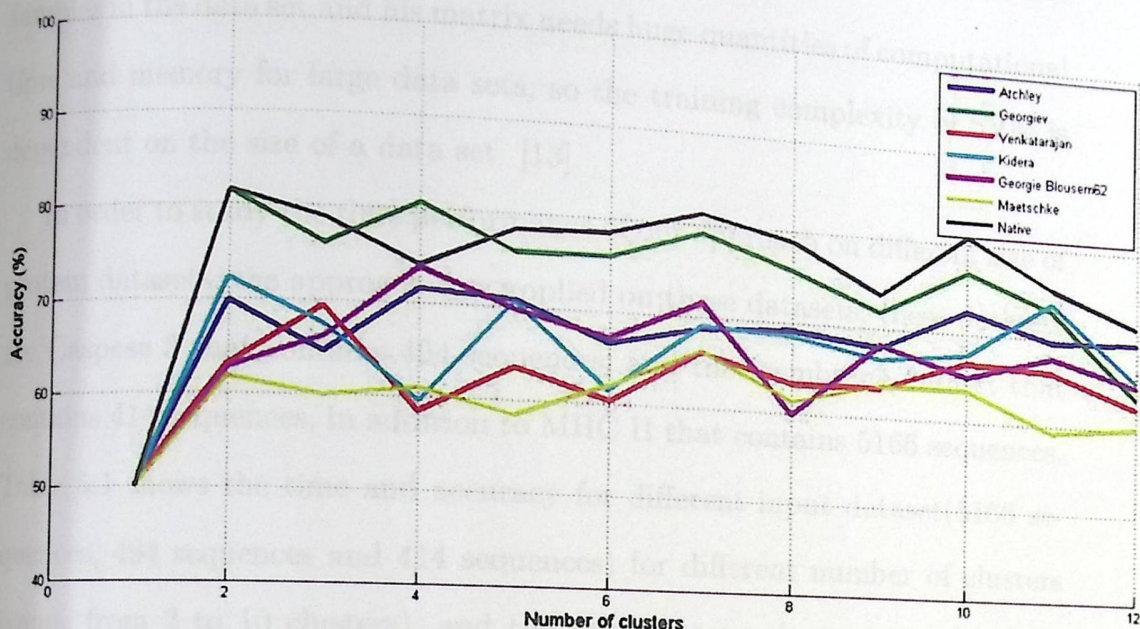


Figure 5.7: Accuracy of SVM for Caspase sequences using concatenating method. The training data divided into different numbers of clusters (range from 2 to 12), one cluster of training data means a classification without clustering.

accuracy results were enhanced approximately by 30% for Georgiev and Native properties (The accuracy reached to 82%).

All sets of properties enhanced the accuracy when using our approach, Georgiev and Native behaved the same, that's because the Georgiev's properties derived from the Native set.

5.4 Time Performance

Many previous studies have focused on reducing the computation time for large data sets by using the clustering before classification, especially when the SVM was used as a classification algorithm because the training time of SVM is a serious obstacle for large data sets, that because training an SVM is usually posed as a quadratic programming (QP) problem to find a hyperplane which implicates a matrix of density $n \times n$, where the n is the number of

5.4. TIME PERFORMANCE

samples in the data set, and his matrix needs huge quantities of computational time and memory for large data sets, so the training complexity of SVM is dependent on the size of a data set [13].

In order to study the time performance of our approach on different size of protein datasets, the approach was applied on three datasets; these datasets are Caspase 3 that contains 494 sequences, and the membrane dataset that contains 414 sequences, in addition to MHC II that contains 5166 sequences. Table 5.1 shows the time and accuracy for different input dataset (5166 sequences, 494 sequences and 414 sequences) for different number of clusters (range from 2 to 10 clusters), and the native properties were used by the PseAAC encoding method.

The specification of our computer that used to run the experiments is as the following: Dell laptop Inspiron 5040, core i5, 8GB RAM.

Table 5.1 shows that the training time for classification is declined when the number of clusters increased for the three datasets, while the accuracy of classification increased or remain in the same rang of accuracy when using the classification without clustering.

The effect of the system on the time can be clearly obvious when the dataset is large as MHC II dataset. Figure 5.8 illustrates the change of the time (in second) with the increase in the number of clusters for the MHC-II data sets because it is the largest dataset compared to the Caspase and membrane protein datasets, it contains 5166 sequences, the native properties were used by the PseAAC encoding method.

When the data divided into two clusters the time decreased from 610 to 97 seconds, the time continued to decline at three clusters, then it began to increase slightly, this increase is due to the overhead caused by increasing the number of clusters.

5.4. TIME PERFORMANCE

Table 5.1: Comparison the time performance and accuracy using the clustering before the classification and without clustering for different size of datasets

Number of sequences	Number of clusters	Accuracy (%)	Time (second)
414	1 (without clustering)	0.502416	0.228455
	2	0.723069	0.032004
	3	0.720195	0.039321
	4	0.763793	0.04904
	5	0.74485	0.05847
	6	0.769374	0.066932
	7	0.767262	0.125557
	8	0.743521	0.090693
	9	0.732575	0.105045
	10	0.746734	0.101628
494	1 (without clustering)	0.546332	0.268538
	2	0.614169	0.06064
	3	0.564923	0.042766
	4	0.565955	0.054031
	5	0.58076	0.07251
	6	0.62807	0.077678
	7	0.603619	0.104219
	8	0.600051	0.099507
	9	0.588401	0.10714
	10	0.605474	0.106795
5166	1 (without clustering)	0.690476	610.2449
	2	0.709833	97.7859
	3	0.698631	5.7150
	4	0.692373	5.8520
	5	0.690527	6.9318
	6	0.688034	7.7595
	7	0.675136	9.1864
	8	0.676268	10.5869
	9	0.675399	11.9592
	10	0.676864	13.6008

This big decline in values is due to the reducing the training data by splitting the training data into clusters, then applying SVM on each cluster that contains fewer training data.

5.4. TIME PERFORMANCE

Table 5.1: Comparison the time performance and accuracy using the clustering before the classification and without clustering for different size of datasets

Number of sequences	Number of clusters	Accuracy (%)	Time (second)
414	1 (without clustering)	0.502416	0.228455
	2	0.723069	0.032004
	3	0.720195	0.039321
	4	0.763793	0.04904
	5	0.74485	0.05847
	6	0.769374	0.066932
	7	0.767262	0.125557
	8	0.743521	0.090693
	9	0.732575	0.105045
	10	0.746734	0.101628
494	1 (without clustering)	0.546332	0.268538
	2	0.614169	0.06064
	3	0.564923	0.042766
	4	0.565955	0.054031
	5	0.58076	0.07251
	6	0.62807	0.077678
	7	0.603619	0.104219
	8	0.600051	0.099507
	9	0.588401	0.10714
	10	0.605474	0.106795
5166	1 (without clustering)	0.690476	610.2449
	2	0.709833	97.7859
	3	0.698631	5.7150
	4	0.692373	5.8520
	5	0.690527	6.9318
	6	0.688034	7.7595
	7	0.675136	9.1864
	8	0.676268	10.5869
	9	0.675399	11.9592
	10	0.676864	13.6008

This big decline in values is due to the reducing the training data by splitting the training data into clusters, then applying SVM on each cluster that contains fewer training data.

5.5. DISCUSSION OF RESULTS

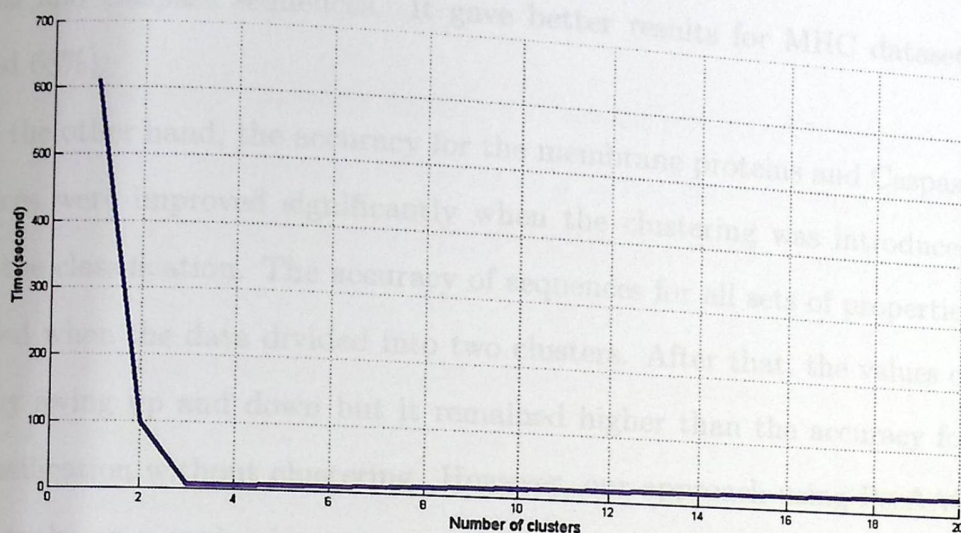


Figure 5.8: The time performance for the proposed approach based on SVM algorithm. The experiment was done using the MHC-II benchmark, the PseAAC encoding method and the Native properties were used

5.5 Discussion of results

This section contains a discussion of the above results, the section is divided into the following subsections: the performance of selected PCPs in our approach, the performance of encoding methods in our approach, the performance of our proposed approach using different benchmarks, and the performance of of SVM computation time using our approach.

5.5.1 Performance of encoding methods in our approach

Two encoding methods were used for all benchmarks (PseAAC and CTD), and an additional method (concatenating method) was used for the Caspase dataset because has fixed length sequences.

PseAAC encoding method

When applying PseAAC to classification without clustering, it was clear that the classification rates were not satisfactory (around 50%) for membrane

5.5. DISCUSSION OF RESULTS

proteins and Caspase sequences. It gave better results for MHC datasets (around 68%).

On the other hand, the accuracy for the membrane proteins and Caspase sequences were improved significantly when the clustering was introduced before the classification. The accuracy of sequences for all sets of properties increased when the data divided into two clusters. After that, the values of accuracy swing up and down but it remained higher than the accuracy for the classification without clustering. However, our approach using PseAAC cannot make an a major improvement for MHC-II sequences. this may be because to the complexity of the MHC-II problem.

CTD encoding method

The CTD of classification without clustering did not give good results for the membrane proteins and Caspase sequences, but the accuracy of the MHC was similar to the results based on PseAAC.

When the clustering was used before the classification, the results were not enhanced in the case of membrane proteins. For the MHC and Caspase it made very little improvement but it not exceed the 4%.

In general, CTD as encoding method did not lead to any enhancement. This is due to nature of CTD method, as mentioned in 2, the CTDs features are not original values of PCPs, instead it is a values derived from dividing the amino acids into groups then depends on the appearing of the groups in the sequences. In addition, CTD depends on the dipeptide features based on division the sequence to pairs.

5.5. DISCUSSION OF RESULTS

Concatenating encoding method

Concatenating method was only applied to Caspase sequences because they have fixed lengths, the results showed that it is not good for classification without clustering, but it is better than PseAAC and CTD in our approach for Caspase, that's because it uses the natural values of the PCPs and that made the differences between the selected sets of properties clear, while almost sets of properties behaved the same based on PseAAC, that's because the PseAAC depends on the features that derived from the natural PCPs, so the values will be close for the same dataset.

Like the PseAAC, the performance of concatenating method depends on properties used to represent the sequences.

5.5.2 Performance of selected descriptors in our approach

In most of the above experiments, we used 7 sets of descriptors (Atchley, Georgive, Venkatarajan, Kidera, Georgive BLOSUM 62, Maetschke and Native). When using PseACC as encoding method, we can notice that the effect of these sets of PCPs on the accuracy for all benchmarks were nearly similar. This is due to the nature of features derived using this encoding method (20 AC + correlation factors), where the first twenty values are similar, so the comparison was mainly done depending on the correlation factors. These correlation factors take into consideration the order of the amino acids in the sequence and the values of the PCPs of amino acids.

On the other hand, when repeating the experiments using the concatenating method, there were clear differences on the classification accuracy among the 7 descriptors. This is because the concatenating method depends

5.5. DISCUSSION OF RESULTS

on the original values of the properties, the results of this method on Caspase sequences showed that Native properties are the best in classification, also the results of Georigive and Native are close, because the Georigive properties were derived from the Native sets themselves.

Based on these results, we can say that the performance of the descriptors in our approach depends on encoding methods that were used.

5.5.3 Performance of our proposed approach on different benchmarks

Based on previous results, our approach that depended on applying the clustering before the classification enhanced the accuracy of classification for two benchmarks; Caspase and membrane proteins, but it failed to improve the accuracy of classification for MHC-II benchmark.

In this thesis we used one clustering algorithm (K-mean) and one classification algorithm (SVM) for all experiments, and we founded that the performance of our approach depended on the encoding methods more than the selected properties, this was clear when we applied two encoding methods; PseAAC and concatenation methods, on the Caspase sequences by using the same sets of properties, the results showed that the concatenating method improved the accuracy by 30%, where the PseAAC improved the accuracy by 15%.

5.5. DISCUSSION OF RESULTS

5.5.4 Performance of SVM training time in our approach

The result of the time on MHC-II showed that our approach significantly reduced the training time of the SVM while maintaining the accuracy of the prediction, and without eliminating any samples.

Conclusion and Future Work

This thesis has proposed a novel approach that aims at enhancing the accuracy of the classification for the protein sequences. This approach based on a clustering algorithm before the classification, using different sets of descriptors based on PCPs, and applying two encoding methods to represent the sequences. The results show that the classification based on the clustering can be significantly enhanced the accuracy of the prediction for the protein sequences, and this enhancement depends on the selected PCPs and the encoding methods used to represent the sequences, this mean that the clusters of the proteins need to examine again to distribute the sequences based on their similarities, in order to facilitate the classification.

This approach has the potential to discover the suitable encoding method and the suitable set of properties for each protein dataset, while the classification without the clustering failed in.

The proposed approach tested on three datasets: membrane proteins, MHC II, Caspase, while the membrane protein dataset represents the full protein sequences, where the MHC and Caspase represent the peptide sequences. Seven sets of descriptors were used to examine the performance of

Chapter 6

Conclusion and Future Work

This thesis has proposed a novel approach that aims at enhancing the accuracy of the classification for the protein sequences. This approach based on the using of clustering algorithm before the classification, using different sets of descriptors based on PCPs, and applying two encoding methods to represent the sequences. The results show that the classification based on the clustering can be significantly enhanced the accuracy of the prediction for the protein sequences, and this enhancement depends on the selected PCPs and the encoding methods used to represent the sequences, this mean that the datasets of the proteins need to examine again to distribute the sequences based on their similarities, in order to facilitate the classification.

This approach has the potential to discover the suitable encoding method and the suitable set of properties for each protein dataset, while the classification without the clustering failed in.

The proposed approach tested on three datasets; membrane proteins, MHC II, Caspase, while the membrane protein dataset represents the full protein sequences where the MHC and Caspase represent the peptide sequences. Seven sets of descriptors were used to examine the performance of

the classification based on clustering, one of them represents original PCPs taken from the amino acids databases, and the rest represent derived PCPs, also two encoding methods; PseAAC and CTD were used to encode the MHC II, membrane protein and the Caspase sequences, and also the concatenation method to encode the Caspase sequences. These encoding methods used to study the influence of the encoding methods on the classification accuracy.

Our experiments show that our approach which depends on using the K-mean clustering before the SVM classifier of the protein sequences can give better results than the classification without the clustering using the selected descriptors and some encoding methods for two datasets; Caspase and the membrane proteins, and it maintains the same range of accuracy for MHC II sequences.

Although our approach could not significantly improve the classification results for MHC II, it succeeds in reducing the training time of the SVM significantly while maintaining the accuracy of prediction. That means our approach can be used to reduce the SVM training time for large datasets, without the need to eliminate any sample from the dataset as in previous approaches.

The results of our experiments show that the PseAAC gave better results than the CTD for the three datasets, and the concatenating method gave better results than PseAAC for Caspase dataset, also the concatenating method was better than other encoding methods to clarify the differences between the selected sets of descriptors.

In general the performance of our approach depends on the descriptors and the encoding methods used to represent the sequences, so to achieve high performance you should find the suitable combination of PCPs and encoding methods.

The main difficulty that we faced in developing our model is the speed limitations, that's because the datasets contain a large number of sequences, especially the MHC II dataset, and for each dataset we need to encode each sequence using the PCPs and cluster the sequences, for our experiments we need to cluster the sequences at least from 2 to 10 clusters, also the classification of the sequences needs time.

In the future other encoding methods, other descriptors can be used to enhance the results of our approach, also different clustering and classification techniques can be used rather than the K-mean and SVM.

The most important development of our approach is to develop a tool depending on this approach in order to help the researcher to know which descriptors, encoding method, clustering and classification algorithms can be used to enhance the accuracy of the prediction for different datasets of proteins.

Based on our approach the reserachers in the future can determine the best descriptors for each dataset (that achieve the higher accuracy).

Bibliography

- [1] T. Abeel, Y. Peer, and Y. Saeys. Java-ml: A machine learning library. *Journal of Machine Learning Research*, 10:931–934, 2009.
- [2] M. Almen, K. Nordström, R. Fredriksson, and H. Schiöth. Mapping the human membrane proteome: a majority of the human membrane proteins can be classified according to function and evolutionary origin. *BMC Biol.*, 2009.
- [3] E. Alpaydm. *Introduction to Machine Learning*. The Adaptive Computation and Machine Learning series, Massachusetts Institute of Technology, 2 edition, 2010.
- [4] S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *statistics Surveys*, 4:4079, 2010.
- [5] W. Atchley, J. Zhao, and A. Fernandes. Solving the protein sequence metric problem. *Proc. Natl. Acad. Sci. USA*, 2005.
- [6] M. Awad, L. Khan, F. Bastani, and I. Yen. An effective support vector machines (svm) performance using hierarchical clustering. *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, pages 663–667, 2004.
- [7] M. Ayyash, H. Tamimi, and Y. Ashhab. Developing a powerful in silico tool for the discovery of novel caspase-3 substrates: a preliminary screening of the human proteome. *BMC Bioinformatics*, 2012.
- [8] A. Ben-Hur, C. Ong, S. Sonnenburg, B. Olkoph, and G. Rtsch. Support vector machines and kernels for computational biology. *PLoS Comput Biol*, 2008.
- [9] K. Bennett and C. Campbell. Support vector machines: Hype or hal-lelujah? *ACM SIGKDD Explorations*, 2, 2000.
- [10] M. Bhasin and G. Raghava. Classification of nuclear receptors based on amino acid composition and dipeptide composition. *J. Bio. Chem.*, 2004.

Bibliography

- [1] T. Abeel, Y. Peer, and Y. Saeys. Java-ml: A machine learning library. *Journal of Machine Learning Research*, 10:931–934, 2009.
- [2] M. Almen, K. Nordström, R. Fredriksson, and H. Schiöth. Mapping the human membrane proteome: a majority of the human membrane proteins can be classified according to function and evolutionary origin. *BMC Biol.*, 2009.
- [3] E. Alpaydm. *Introduction to Machine Learning*. The Adaptive Computation and Machine Learning series, Massachusetts Institute of Technology, 2 edition, 2010.
- [4] S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *statistics Surveys*, 4:4079, 2010.
- [5] W. Atchley, J. Zhao, and A. Fernandes. Solving the protein sequence metric problem. *Proc. Natl. Acad. Sci. USA*, 2005.
- [6] M. Awad, L. Khan, F. Bastani, and I. Yen. An effective support vector machines (svm) performance using hierarchical clustering. *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, pages 663–667, 2004.
- [7] M. Ayyash, H. Tamimi, and Y. Ashhab. Developing a powerful in silico tool for the discovery of novel caspase-3 substrates: a preliminary screening of the human proteome. *BMC Bioinformatics*, 2012.
- [8] A. Ben-Hur, C. Ong, S. Sonnenburg, B. Olkoph, and G. Rtsch. Support vector machines and kernels for computational biology. *PLoS Comput Biol*, 2008.
- [9] K. Bennett and C. Campbell. Support vector machines: Hype or hal-lelujah? *ACM SIGKDD Explorations*, 2, 2000.
- [10] M. Bhasin and G. Raghava. Classification of nuclear receptors based on amino acid composition and dipeptide composition. *J. Bio. Chem.*, 2004.

BIBLIOGRAPHY

- [11] K. Brodersen, C. Ong, K. Stephany, and J. Buhmann. The balanced accuracy and its posterior distribution. *International Conference on Pattern Recognition*, 2010.
- [12] C. Burges. Dimension reduction: A guided tour. *Foundations and Trends in Machine Learning*, 2:275365, 2009.
- [13] J. Cervantes, X. Li, and W. Yu. Support vector machine classification based on fuzzy clustering for large data sets. *MICAI'06 Proceedings of the 5th Mexican international conference on Artificial Intelligence*, pages 572–582, 2006.
- [14] C. Chou. Prediction of protein subcellular locations by incorporating quasi sequence-order effect. *Biochem Biophys Res Communication*, pages 477–483, 2000.
- [15] C. Chou. Prediction of protein cellular attributes using pseudo-amino-acid composition. *PROTEINS: Structure, Function, and Genetic*, pages 246–255, 2001.
- [16] P. Cunningham. Dimension reduction. *University College Dublin Technical Report UCD-CSI-2007-7*, 2007.
- [17] I. Dubchak, I. Muchink, S. Holbrook, , and S. Kim. Prediction of protein folding class using global description of amino acid sequence. *Proc.Natl.Acad.Sci.USA*, pages 8700–8704, 1995.
- [18] D. Eisenberg, R. Weiss, T. Terwilliger, and W. Wilcox. Hydrophobic moments and protein structure. *Faraday Symp. Chem. Soc.*, 17:109–120, 1982.
- [19] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27:861874, 2006.
- [20] I. Fodor. A survey of dimension reduction techniques. *Lawrence Livermore National Laboratory , Technical Information Departments Digital Library*, 2002.
- [21] Q. Gao, Ye X., Jin Z., and J. He. Improving discrimination of outer membrane proteins by fusing different forms of pseudo amino acid composition. *Analytical Biochemistry*, 398:5259, 2010.
- [22] A. Georgiev. Interpretable numerical descriptors of amino acid space. *JOURNAL OF COMPUTATIONAL BIOLOGY*, 16(5), 2009.
- [23] S. Gunn. Support vector machines for classification and regression. *Technical Report*, 1998.

BIBLIOGRAPHY

- [24] S. Hellberg, M. Sjostrom, and S. Wold. The prediction of bradykinin potentiating potency of pentapeptides. an example of a peptide quantitative structureactivity relationship. *Acta Chem. Scand.*, pages 135–140, 1986.
- [25] A. Hertzmann and D. Fleet. Machine learning and data mining. *University of Toronto*, 2010.
- [26] F. Hosseinzadeh, M. Ebrahimi, B. Goliaei, and N. Shamabadi. Classification of lung cancer tumors based on structural and physicochemical properties of proteins by bioinformatics models. *PLoS ONE*, 7(7), 2012.
- [27] C. Immunological Bioinformatics. Center for biological sequence analysis. <http://www.cbs.dtu.dk/suppl/immunology/NetMHCII-2.0.php>.
- [28] O. James and U. Emmanuel. Comparative studies on the protein and mineral composition of some selected nigerian vegetables. *African Journal of Food Science*, 5(1):22–25, 2011.
- [29] S. Junior. Sequence alignment algorithms. *M.Sc. in Advanced Computing, Utrecht University Kings College London*, 2003.
- [30] S. Kawashima and M. Kanehisa. Aaindex: Amino acid index database. *Nucleic Acids Research*, 27:27–36, 1999.
- [31] C. Kesmir. Bioinformatics. *Utrecht University*, 2013.
- [32] A. Kidera, Y. Konishi, and M. Oka. Statistical analysis of the physical properties of the 20 naturallyoccurring amino acids. *J. Protein Chem.*, 4:2355, 1985.
- [33] A. Kyriakopoulou and T. Kalamboukis. Text classification using clustering. In *Proceedings of the ECML-PKDD Discovery Challenge Workshop*, 2006.
- [34] M. Lapinsh, A. Gutcaits, P. Prusis, C. Post, T. Lundstedt, and Wikberg J. Classification of g-protein coupled receptors by alignment-independent extraction of principal chemical properties of primary amino acid sequences. *Protein Science*, 2002.
- [35] L. Maaten. An introduction to dimensionality reduction using matlab. *Universiteit Maastricht*, 2007.
- [36] S. Maetschke, M. Towsey, and M. Boden. Blomap: an encoding of amino acids with improves signal peptide cleavage site prediction. *Proceedings Third Asia Pacific Bioinformatics Conference*, 7:141–150, 2005.
- [37] V. Mathura and D. Kolippakkam. Apdbase: Amino acid physicochemical properties database. *Bioinformatics*, 1, 2005.

BIBLIOGRAPHY

- [38] The Mathworks. Statistical toolbox 7.0. <http://www.mathworks.com/help/stats/index.html>.
- [39] M. McKee and J. McKee. *Biochemistry: The Molecular Basis of Life*. Oxford University Press, USA, 5 edition, 2011.
- [40] R. Michalski, J. Carbonell, and T. Mitchell. Machine learning: An artificial intelligence approach. *Tioga Publishing Company*, 1983.
- [41] X. Nan, D. Cao, Q. Xu, and Y. Liang. protr: Protein sequence feature extraction with r. *Central South University*, 2012.
- [42] L. Nanni, S. Brahnam, and A. Lumini. High performance set of pseAAC and sequence based descriptors for protein classification. *Journal of Theoretical Biology*, 2010.
- [43] L. Nanni and A. Lumini. A new encoding technique for peptide classification. *Expert Systems with Applications*, 38:3185–3191, 2011.
- [44] S. Ong, H. Lin, Y. Chen, Z. Li, and Z. Cao. Efficacy of different protein descriptors in predicting protein functional families. *BMC Bioinformatics*, 2007.
- [45] S. Opiyo and E. Moriyama. Protein family classification with partial least squares. *Proteome Res.*, 6:846853, 2007.
- [46] K. Park, Gromiha M., P. Horton, and M. Suwa. Discrimination of outer membrane proteins using support vector machines. *Bioinformatics*, 21:223–229, 2005.
- [47] A. Prlic, A. Yates, and et al. Bliven, S. Biojava: an open-source framework for bioinformatics. *Bioinformatics*, 28:26932695, 2012.
- [48] S. Rackovsky. Sequence physical properties encode the global organization of protein structure space. *Proceedings of the National Academy of Sciences of the United States of America*, 2009.
- [49] A. Rahideh and M. Shaheed. Cancer classification using clustering based gene selection and artificial neural networks. *2nd International Conference on Control, Instrumentation and Automation (ICCIA)*, 2011.
- [50] M. Rajapakse, B. Schmidt, L. Feng, and V. Brusic. Predicting peptides binding to mhc class II molecules using multi-objective evolutionary algorithms. *BMC Bioinformatics*, 2007.
- [51] S. Ray and T. Kepler. Amino acid biophysical properties in the statistical prediction of peptide-mhc class I binding. *Immunome Research*, 2007.

BIBLIOGRAPHY

- [52] S. Ray and T. Kepler. Amino acid biophysical properties in the statistical prediction of peptide-mhc class i binding. *Immunome Research*, 2007.
- [53] R. Saidi, M. Maddouri, and E. Nguifo. Protein sequences classification by means of feature extraction with substitution matrices. *BMC Bioinformatics*, 2010.
- [54] M. Sandberg, L. Eriksson, J. Jonsson, and et al. New chemical descriptors relevant for the design of biologically active peptides. a multivariate characterization of 87 amino acids. *J. Med. Chem.*, page 24812491, 1998.
- [55] P. Sneath. Relations between chemical structure and biological activity in peptides. *Journal of Theoretical Biology*, 12:157 195, 1996.
- [56] P. Tan, M. Steinbach, and Kumar V. Introduction to data mining. *Addison-Wesley Companion*, 2006.
- [57] K. Tomii and M. Kanehisa. Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins. *Protein Engineering*, 9:27–36, 1996.
- [58] L. Tussey and A. McMichael. General introduction to the mhc. *Cambridge University Press*, 1995.
- [59] M. Venkatarajan and W. Braun. New quantitative descriptors of amino acids based on multidimensional scaling of a large number of physical-chemical properties. *J. Mol. Modeling*, 7:445453, 2001.
- [60] J. Vesanto and E. Alhoniemi. Clustering of the self-organizing map. *IEEE*, 11, 2000.
- [61] G. Wardlaw and A. Smith. *Contemporary Nutrition*. McGraw-Hill Science, Engineering, Math, 6 edition, 2006.
- [62] Y. Xiong, J. Liu, W. Zhang, and T. Zeng. Prediction of heme binding residues from protein sequences with integrative sequence profiles. *Proteome Science*, 10, 2012.
- [63] H. Yu, J. Yang, and J. Han. Classifying large data sets using svms with hierarchical clusters. *ACM, Knowledge Discovery and Data Mining conference*, 2003.

BIBLIOGRAPHY

- [52] S. Ray and T. Kepler. Amino acid biophysical properties in the statistical prediction of peptide-mhc class i binding. *Immunome Research*, 2007.
- [53] R. Saidi, M. Maddouri, and E. Nguifo. Protein sequences classification by means of feature extraction with substitution matrices. *BMC Bioinformatics*, 2010.
- [54] M. Sandberg, L. Eriksson, J. Jonsson, and et al. New chemical descriptors relevant for the design of biologically active peptides. a multivariate characterization of 87 amino acids. *J. Med. Chem.*, page 24812491, 1998.
- [55] P. Sneath. Relations between chemical structure and biological activity in peptides. *Journal of Theoretical Biology*, 12:157 195, 1996.
- [56] P. Tan, M. Steinbach, and Kumar V. Introduction to data mining. *Addison-Wesley Companion*, 2006.
- [57] K. Tomii and M. Kanehisa. Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins. *Protein Engineering*, 9:27–36, 1996.
- [58] L. Tussey and A. McMichael. General introduction to the mhc. *Cambridge University Press*, 1995.
- [59] M. Venkatarajan and W. Braun. New quantitative descriptors of amino acids based on multidimensional scaling of a large number of physical-chemical properties. *J. Mol. Modeling*, 7:445453, 2001.
- [60] J. Vesanto and E. Alhoniemi. Clustering of the self-organizing map. *IEEE*, 11, 2000.
- [61] G. Wardlaw and A. Smith. *Contemporary Nutrition*. McGraw-Hill Science, Engineering, Math, 6 edition, 2006.
- [62] Y. Xiong, J. Liu, W. Zhang, and T. Zeng. Prediction of heme binding residues from protein sequences with integrative sequence profiles. *Proteome Science*, 10, 2012.
- [63] H. Yu, J. Yang, and J. Han. Classifying large data sets using svms with hierarchical clusters. *ACM, Knowledge Discovery and Data Mining conference*, 2003.