COMPUTATIONAL REPRESENTATION OF PROTEIN SEQUENCES FOR HOMOLOGY DETECTION AND CLASSIFICATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HASAN OĞUL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE DEPARTMENT OF INFORMATION SYSTEMS

JANUARY 2006

Approval of the Graduate School of Informatics

Assoc. Prof. Dr. Nazife BAYKAL
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

Assoc. Prof. Dr. Onur DEMİRÖRS
Head of Department

This is to certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Assist. Prof. Dr. Erkan Ü. MUMCUOĞLU

Supervisor

Examining Committee Members

Prof. Dr. Volkan ATALAY	(METU,CENG)	
Assist. Prof. Dr. Erkan Ü. MUMCUOĞLU	(METU,II)	
Assoc. Prof. Dr. Nazife BAYKAL	(METU,II)	
Assist. Prof. Dr. Özlen KONU	(Bilkent Univ., MBG)	
Assoc. Prof. Dr. Yasemin YARDIMCI	(METU,II)	

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name and Surname: Hasan OĞUL

Signature:

ABSTRACT

COMPUTATIONAL REPRESENTATION OF PROTEIN SEQUENCES FOR HOMOLOGY DETECTION AND CLASSIFICATION

Oğul, Hasan
Ph.D., Department of Information Systems
Supervisor: Assist. Prof. Dr. Erkan Ü. MUMCUOĞLU

January 2006, 102 pages

Machine learning techniques have been widely used for classification problems in computational biology. They require that the input must be a collection of fixed-length feature vectors. Since proteins are of varying lengths, there is a need for a means of representing protein sequences by a fixed-number of features. This thesis introduces three novel methods for this purpose: *n*-peptide compositions with reduced alphabets, pairwise similarity scores by maximal unique matches, and pairwise similarity scores by probabilistic suffix trees.

New sequence representations described in the thesis are applied on three challenging problems of computational biology: remote homology detection, subcellular localization prediction, and solvent accessibility prediction, with some problem-specific modifications. Rigorous experiments are conducted on common benchmarking datasets, and a comparative analysis is performed between the new methods and the existing ones for each problem.

On remote homology detection tests, all three methods achieve competitive accuracies with the state-of-the-art methods, while being much more efficient. A combination of new representations are used to devise a hybrid system, called PredLOC, for predicting subcellular localization of proteins and it is tested on two distinct eukaryotic datasets. To the best of author's knowledge, the accuracy achieved by PredLOC is the highest one ever reported on those datasets. The maximal unique match method is resulted with only a slight improvement in solvent accessibility predictions.

Keywords: *n*-peptide composition, maximal unique match, probabilistic suffix tree, remote homology, subcellular localization.

ÖZ

PROTEİN DİZİLİMLERİNİN HOMOLOJİ SEZİMİ VE SINIFLANDIRMA AMAÇLI BİLİŞİMSEL GÖSTERİMİ

Oğul, Hasan
Doktora, Bilişim Sistemleri A.B.D
Tez Yöneticisi: Yrd. Doç. Dr. Erkan Ü. MUMCUOĞLU

Ocak 2006, 102 sayfa

Otomatik öğrenme yöntemleri bilişimsel biyolojide sınıflandırma problemleri için sıkça kullanılmaktadır. Bu yöntemlerin girdilerinin sabit uzunlukta özellik vektörlerinden oluşması gerekir. Proteinler farklı uzunluklarda olabileceği için, protein dizilimlerini sabit sayıdaki özelliklerle temsil edecek yöntemlere ihtiyaç duyulmaktadır. Bu tezde bu amaçla üç farklı yöntem sunulmaktadır. Bunlardan birincisi azaltıltılmış alfabelerle *n*-peptid bileşimi, ikincisi en büyük benzersiz eşleşmelere göre ikili benzerlik değerleri, ve üçüncüsü ise olasılıksal sonek ağaçları ile ikili benzerlik değerleridir.

Tezde tarif edilen yeni dizilim gösterim yöntemleri, probleme özgü değişiklilerle birlikte, bilişimsel biyolojinin üç önemli problemi üzerinde uygulanmıştır; uzak homoloji sezimi, hücresel konumlanma tahmini, çözgen erişebilirlik tahmini. Her problem için, ortak kıyaslama kümeleri üzerinde yapılan deneyler sonucunda, mevcut yöntemlerle yeni yöntemler arasında karşılaştırma analizleri sunulmuştur.

Uzak homoloji sezimi testlerinde, üç yeni yöntemin hepsi mevcut en iyi yöntemlerle karşılaştırılabilir doğruluk değerleri elde ederken, bunların çok daha verimli çalıştıkları gözlenmiştir. Yeni yöntemlerin bir kombinasyonu, proteinlerin hücresel konumlanmalarını tahmin eden PredLOC isimli sistemi geliştirmek için kullanılmış ve bu sistem iki farklı ökaryotik protein kümesi için test edilmiştir. PredLOC her iki veri kümesi için de şu ana kadar elde edilen en iyi doğruluk değerine ulaşmıştır. En büyük benzersiz eşleşmelerin kullanımı, çözgen erişebilirlik tahmininde az miktarda iyileştirme sağlayabilmiştir.

Anahtar kelimeler: *n*-peptid bileşimi, en büyük benzersiz eşleşme, olasılıksal sonek ağacı, uzak homoloji, hücresel konumlanma.

To white butterfly…

ACKNOWLEDGMENTS

I would like to state my special thanks to many people who provided their sincere guidance, support and motivations at several stages of my thesis;

Assist. Prof. Dr. Erkan Mumcuoğlu (my supervisor): Thank you for providing a friendly working environment and significant guidance at all phases of the thesis.

Prof. Dr. Mahinur Akkaya and Assist. Prof. Dr. Özlen Konu (bioscientists): I certainly could not have completed this thesis without your "biohelp" that you provided either in progress meetings or at any time you were available.

Prof. Dr. İmdat Kara (my dean in B.Ü): Thank you for motivations, supports and attempts for making my studies easier in terms of my workload.

Prof. Dr. Hayri Sever: The short discussions that we made in your office have been valuable in accomplishing my thesis. Thank you for your supports and guidance.

Emre, Pelin, Tolga, Halit and other co-workers: Thank you for your help and assistance throughout the thesis.

Mehmet and Rahime Oğul (my parents): I am proud to be your son. Thank you for your affection and encouragement.

Emine Oğul (my wife): You have been my main source of motivation at any time that I fallen in desperation. Thank you for your endless patience and tolerance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BLAST: Basic Local Alignment Search Tool

BLOSUM: Block Substitution Matrix

CATH: Database of Class, Architecture, Topology and Homology

DNA: Deoxyribonucleic Acid

DSSP: Dictionary of Secondary Structure of Proteins

FASTA: Fast Alignment

HMM: Hidden Markov Model

MUM: Maximal Unique Match

kNN: k-nearest Neighborhood

NCBI: National Center for Biotechnology Information

NN: Neural Network

PAM: Point Accepted Mutation

PDB: Protein Data Bank

PST: Probabilistic Suffix Tree

RNA: Ribonucleic Acid

ROC: Receiver Operating Characteristic

SCOP: Structural Classification of Proteins

SVM: Support Vector Machine

CHAPTER 1

Introduction

High-throughput genome sequencing projects have been resulted with the accumulation of a large amount of raw sequence data in many databases. Owing to the experimental complications and obstacles in the structural and functional analysis of proteins, the amount of discrepancy between the number of known protein sequences and the number of experimentally determined structures has steadily increased in recent years. This situation has given occasion to the emergence of computational tools and methodologies that make automated annotations on structure and function of proteins using the sequence information available in public databases.

1.1 Scope and Organization of the Thesis

This thesis deals with developing accurate and efficient representations for protein sequences to be used for automated protein classification systems. Throughout the thesis, three novel sequence representation schemes are introduced and their applicability on various protein-related problems is discussed. Besides the generic representations that suit many applications, several problem-specific modification strategies are proposed. The new representations are applied on three challenging problems of computational biology and experimental results are presented.

The next section of this chapter follows with the introduction of some biological concepts and definitions that are frequently referred and related at several parts of the thesis. This chapter ends with the introduction of some publicly available databases that were used to create the data sets on which the experiments were carried out.

The second chapter is where the new sequence representations are described. The first representation with *n*-peptide compositions is defined in Section 2.1. Similarity-based representations are described in Section 2.2. Among them, the maximal unique match model is described in Section 2.2.2 and the pairwise probabilistic suffix tree model is described in Section 2.2.3. Although alignment-based sequence comparison techniques are not original to this study, they are also introduced in Section 2.2.1 to give a historical perspective in sequence analysis.

The next chapters deal with the application of new representations on three important biological problems; remote homology detection (Chapter 3), subcellular localization prediction (Chapter 4) and residue solvent accessibility prediction (Chapter 5). For all applications, rigorous analyses are presented as a result of experiments conducted on common benchmarking data sets.

1.2 Contributions

The thesis introduces novel approaches for protein sequence analysis. Moreover, it provides significantly improved solutions for some biologically important problems. The contributions of the thesis can be summarized as follows:

- *n*-peptide compositions are made usable for *n*>3 with gradually simplified amino acid alphabets. This provides both the advantage of reduced complexity and the opportunity to evaluate evolutionarily possible mismatches which has never considered in compositional representations in previous studies.

- The maximal unique match definition is used for the first time for protein sequences to infer remote homologies and shown to be very useful in spite of its simplicity and efficiency.
- This study is the first use of probabilistic suffix trees for pairwise sequence comparison. With some modifications, pairwise probabilistic suffix tree model is shown to be better than original family-based probabilistic suffix tree model when integrated into a discriminative framework for protein classification.
- New methods provide more efficient solution to discriminative remote homology detection problem, while their accuracies are comparable to that of the state-of-the-art methods.
- A new system for predicting subcellular localization of eukaryotic proteins is designed using new sequence representations and its Linux-compatible source codes and binaries are made freely available for academic users. In the experiments, the new system provided the best accuracy ever reported on the benchmark datasets.
- Solvent accessibility problem is revisited. In addition to a simple and slightly improved residue representation scheme, a prediction refinement strategy is proposed based on the maximal unique matches.

1.3 Biological Background

The genetic information which we inherit from our parents and passes to our children is carried by a long molecule called as deoxyribonucleic acid or DNA. DNA has two long strands, each of which is made up from chemical units called phosphates, deoxyribose sugars, and nucleotides linked as a sequence. The nucleotides in DNA are of four kinds; Adenine, Guanine, Cytosine, and Thymine, and they are abbreviated by A, G, C, T, respectively. The DNA molecule is composition of these two strands which lie in an antiparallel orientation to form a double helix. This conformation follows strict base pairing rules such that A can only make a pair with T and G can only make a pair with C. Thus, each strand is actually a complementary sequence of other strand. One of the strands in DNA helix is called as template and the other is called as coding strand. Figure 1 shows a simplified view of a DNA molecule.

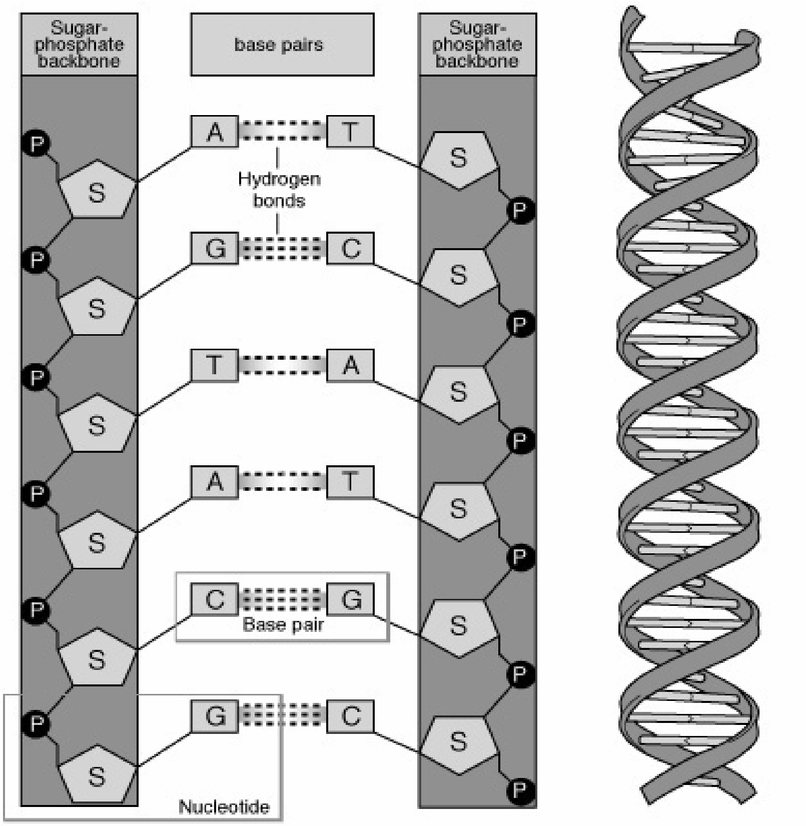


Figure 1. Two different views of a DNA molecule.*

Many of the chemical reactions in a cell are mainly the result of proteins. The basic role of DNA in an organism is to control the activities in cells by specifying the synthesis of proteins. However, a DNA does not directly produce a protein; instead it generates a template in the form of a strand of RNA, which in turn codes the protein synthesis. In general, the information flows from DNA to proteins through RNA molecules.

* The figure was obtained from http://www.accessexcellence.org/AB/GG/dna2.html

The generation of template RNA from DNA is called *transcription*. Transcription is the mechanism by which a template strand of DNA is utilized to form one the three different classifications of RNA. These classes are as follows:

1. mRNAs are the genetic coding templates used by translational machinery to determine the order of amino acids incorporated into protein polypeptide chain.
2. tRNAs are small structures used to recognize the encoded sequence of mRNAs to allow correct insertion of amino acids into the elongating polypeptide chain.
3. rRNAs are assembled, together with numerous ribosomal proteins, to form a catalytic domain into which the tRNAs enter with their attached amino acids. The ribosomes catalyze all the functions of protein synthesis.

After transcription, the resulting RNA is complementary to the template strand of the DNA duplex and identical to the non-template strand. The non-template strand of DNA refers to coding strand because all the sequence in this strand would then be translated into proteins through mRNA. However, urasil (U) is substituted for timing (T) in RNA. The synthesis of protein is directed by RNAs and the process is called as *translation*. This process requires all three classes of RNA, however, the template for correct addition of individual amino acids is the mRNA. In eukaryotes, mRNA sequence is exported out of the nucleus to the cytoplasm for translation into a protein primary sequence. The RNA then translated as a series of three-letter codons, where each codon represents a particular amino acid. The list of amino acids and the abbreviations used are given in Table 1.

Although 64 possible combinations of the 3 nucleotides code for amino acids are available, the code is degenerate since there are only 20 amino acids in nature. All possible codon sequences and corresponding amino acid representatives are give in Table 2. It is experimentally shown that some amino acids are encoded by more than one triplet codon, hence the degeneracy of the genetic code. A special codon called as *start codon* signals the starting of point of translation. The process of translation terminates with another special codon called as *stop codon*.

Table 1. Amino acids and their abbreviations.

Nonpolar Amino Acids (hydrophobic)		
amino acid	three letter code	Single letter code
Glycine	Gly	G
Alanine	Ala	A
Valine	Val	V
Leucine	Leu	L
Isoleucine	Ile	I
Methionine	Met	M
Phenylalanine	Phe	F
Tryptophan	Trp	W
Proline	Pro	P
Polar Amino Acids (hydrophilic)		
Serine	Ser	S
Threonine	Thr	T
Cysteine	Cys	C
Tyrosine	Tyr	Y
Asparagine	Asn	N
Glutamine	Gln	Q
Electrically Charged Amino Acids (negative and hydrophilic)		
Aspartic acid	Asp	D
Glutamic acid	Glu	E
Electrically Charged Amino Acids (positive and hydrophilic)		
Lysine	Lys	K
Arginine	Arg	R
Histidine	His	H

To summarize the process, the information-storage molecule (DNA) transfers its information through a transfer molecule (RNA) to a functional, noncoding product (protein). For example; if the DNA template sequence is as;

AGTAATCTCGTTACT,

then the RNA sequence will be same as DNA template sequence, except that all **T**s are replaced by **U**s;

AGU AAU CUC GUU ACU.

And the protein sequence would be as the sequence of corresponding amino acid abbreviations for the triplet codons in RNA;

S N L V T.

Table 2. The genetic code.

	Middle				
First	U	C	A	G	Last
U	Phe	Ser	Tyr	Cys	U
	Phe	Ser	Tyr	Cys	C
	Leu	Ser	Stop	Stop	A
	Leu	Ser	Stop	Trp	G
C	Leu	Pro	His	Arg	U
	Leu	Pro	His	Arg	C
	Leu	Pro	Gln	Arg	A
	Leu	Pro	Gln	Arg	G
A	Ile	Thr	Asn	Ser	U
	Ile	Thr	Asn	Ser	C
	Ile	Thr	Lys	Arg	A
	Met	Thr	Lys	Arg	G
G	Val	Ala	Asp	Gly	U
	Val	Ala	Asp	Gly	C
	Val	Ala	Glu	Gly	A
	Val	Ala	Glu	Gly	G

The structure of a protein is visualized in four levels. The primary structure of a protein refers to the linear number and order of amino acids present in the polypeptide chain. The convention for the designation of the order of amino acids is from the N-terminal end, i.e. the end residue with the free amino group, to the C-terminal end, i.e. the end residue with carboxyl group.

The ordered array of amino acids in a protein confers regular conformational forms upon that protein. These conformations constitute the secondary structure of a protein. In general, proteins fold into two broad class of structure termed as globular proteins or fibrous proteins. Globular proteins are compactly folded and coiled, whereas, fibrous proteins are more filamentous and elongated. It is the partial double-bond character of the peptide bond that defines the conformations a protein chain may assume. Within a single protein, different regions of the polypeptide chain may assume different conformations determined by the primary sequence of the amino acids.

The common secondary structure elements are α-helices, extended β-sheets, bends, bridges, turns and loops. Of these, α-helices and β-sheets are more definite structure elements. The others are not so easily described and, most commonly, they all are called as "others" or "coils".

The α-helix is encountered in proteins of globular class. The formation of α-helix is spontaneous and is stabilized by H-bonding between amide nitrogens and carbonyl carbons of peptide bonds spaced four residues apart. Whereas an α-helix is composed of a single linear array of helically disposed amino acids, β-sheets are composed of 2 or more different regions of stretches of at least 5-10 amino acids.

Tertiary structure refers to the complete three dimensional structure of the polypeptide units of a given protein. Included in this description is the spatial relationship of different secondary structures to one another within a polypeptide chain and how these secondary structures themselves fold into the three-dimensional form of the proteins. Secondary structure of proteins often constitutes distinct domains. Therefore, tertiary structure also describes the relationship of different domains to one another within a protein. The interactions of different domains is governed by several forces, including H-bonding, hydrophobic interactions, electrostatic interactions and Van Der Waals forces. Figure 2 shows an example tertiary structure on which the local secondary structure elements are depicted by different shapes.

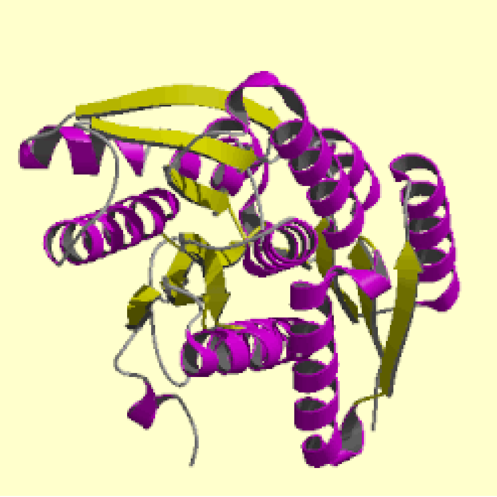


Figure 2. The view of protein 2gpr (from CATH database).

Proteins may contain 2 or more different polypeptide chains that are held in association by the same non-covalent forces that stabilize the tertiary structure of proteins. The structure formed by more than one polypeptide chains is known as quaternary structure.

The higher level structures of a protein are important to identify the function, activity or environmental interactions of that protein. The functions of proteins are the essence of life itself. Many of the proteins in an organism are enzymes, specific proteins that speed up the rate of chemical reactions in the cell. Enzymes are tiny molecular tools that temporarily combine with the ingredients for a specific reaction and hold them at the correct angle for a reaction to occur. They also lower the amount of energy needed for reaction to proceed so it can occur at a much lower temperature than would be necessary without enzyme. In addition, proteins play many other roles in the cell. Proteins may be classified with respect to those activities realized in the cell; some categories are enzymes, structural proteins, hormones, transport proteins, etc. Each type of proteins may further be categorized according to their functions. For instance, transport proteins may further be referred as oxygen transporters or fatty acid transporters, etc.

It is already stated that the structure determines the function of a protein. However, the next, maybe the more important, question is what determines the structure of a protein. It is shown for many proteins that the primary structure (amino acid sequence) of a protein is major determinant for the tertiary structure (Anfinsen, 1973). However, knowledge of the primary structure is not sufficient; the native solution environment also plays a role in the three-dimensional conformation of a protein.

1.4 Source of Data

There are many databases to store and efficiently retrieve the biological data. Whereas some databases are specialized to store a specific kind of information, the others are of general purpose and serve many facilities to search much kind of data. Many of the databases can be accessed online via their web sites. Almost all of them are arranged so that you can download their data into your local workstations. Since proteins are our concern, some of the important protein-related databases are listed and briefly introduced in Appendices.

Throughout the thesis, many of those databases have been frequently accessed either to obtain experimental data or to verify the results. Protein family and superfamily definitions are obtained from SCOP database (Murzin et al. 1995) for remote homology detection tests. Protein sequences were downloaded from either PDB (Berman et al., 2000) or SWISSPROT (Bairoch and Apweiler, 1999) in fasta format. The subcellular localizations were annotated from SWISSPROT database. The relative solvent accessibility values were obtained from DSSP (Kabsch and Sander, 1983) and reorganized. In addition to those databases which were used as the source of experimental data, CATH (Orengo, Michie, Jones, Jones, Swindells and Thornton, 1997), NCBI and Pfam (Bateman et al., 2002) databases and their tools were frequently used to check the correctness and integrity of the data used in the experiments.

CHAPTER 2

Protein Sequence Representations

Automated categorization of proteins into their structural or functional classes is an important challenge for computational biology. Most of the recent methods that have been developed to assign proteins into well-known classes are based on the machine learning techniques such as neural networks, Bayesian classifiers and support vector machines. The classification theory suggests that the input to a classifier must be a collection of fixed-length feature vectors. Both the training and testing samples are required to satisfy this condition. Since a protein is an ordered chain of varying number of amino acids, the sequence itself is not a consistent data such that it can not be directly fed into a machine learning classifiers. Therefore, there is a need for a means of representing proteins by a fixed-length feature vector.

2.1 Compositional Representation

Many methods have been proposed for protein sequence representation. Amino acid composition of a protein may be the most widely used method to represent protein sequences. In this scheme, each protein is represented by a 20 dimensional feature vector where each dimension is the fraction of corresponding amino acid in the sequence. In spite of its basic definition, amino acid composition has been successfully applied in many problems (Bahar, Atilgan, Jernigan, Erman, 1997; Zhang, Chou and Maggiora, 1995; Reinhardt and Hubbard, 1998; Hua and Sun, 2001) and still considered to be a sufficient knowledge in the detection of structural classes such as all-alpha, all-beta or alpha+beta proteins. The main problem with this scheme is that it ignores the local order of amino acid along the sequence. To include the effect of local amino acid orders, dipeptide compositions have been utilized as a single representation or in a combination vector with amino acid composition. Dipeptide composition is a vector having 400 dimensions each of which represents the fraction of one of the possible 2-length amino acid strings. Dipeptide composition has also been used in many problems (Nakashima and Nishikawa, 1994; Park and Kanehisa, 2003; Jin, Weiwu and Tang, 2003; Yu, Lin, Hwang, 2004) and shown to perform better than amino acid composition in most cases. However, it still lacks of the information about the order of amino acids in longer peptides.

Another protein representation scheme built on the sequence content is based on the use of known physicochemical properties of proteins. Ding and Dubchak (2001) used six properties: amino acids, predicted secondary structure, hydrophobicity, normalized van der Waal volume, polarity and polarizability. For each property, they extracted features from primary sequence based on three descriptors; percent composition of three constituents (e.g. polar, neutral and hydrophobic residues in hyrophobicity), the transition frequencies (polar to neutral, neutral to hydrophobic, etc) and the distribution patterns of constituents. They applied this scheme for multi-class protein fold recognition with support vector machines and neural networks. Similar representation was used for protein function classification via support vector machines (Cai, Wang, Sun, Chen, 2003). Bashin and Raghava (2004) used 33 different physico-chemical properties and built an input vector having 33 scalar values, each representing the average value of a distinct property contained in the protein to predict subcellular localizations.

Motif-based protein representations are also related to the sequence content. In these methods, instead of searching all possible amino acid or dipeptide combinations, some previously annotated and structurally known or deposited to be important sequence patterns, which are available in public databases such as PROSITE (Falquet et al., 2002), I-SITES (Bystroff and Baker, 1998), BLOCKS (Henikoff, Henikoff, Pietrokovski, 1999), eMOTIF (Huang and Brutlag, 2001), are searched in the sequence and a boolean value that indicates the existence of the corresponding motif inside the sequence is associated with each dimension of the feature vector (Ben-hur and Brutlag, 2003; Hou, Hsu, Lee, Bystroff, 2003). In this case, the number of dimensions in the feature vector becomes equal to the number of available motifs in the database used. In spite of their remarkable success, motif-based approaches has the disadvantage that the resulting feature vectors often contain too many sparse data, since many of the given motifs are not necessarily to be included in the target sequence.

In this study, proteins are represented by their *n*-peptide compositions, which is a generalization of amino acid and dipeptide compositions. For each value of *n*, corresponding feature vector contains the fraction of each possible *n*-length substring in the sequence. For example, the feature vector refers to amino acid composition for $n=1$, and dipeptide composition for $n=2$. As mentioned above, *n*-peptide compositions have been used extensively for many problems with $n=1$ and $n=2$, however, composition of longer *n*-peptides could not be used efficiently since the time and memory requirements exponentially increase with the value of *n*. The number of dimensions in the feature vector corresponding to *n*-peptide composition is 20^n. The memory space complexity of the training step then becomes $O(k20^n)$, where *k* is the number of proteins in the training set. This leads to formation of high-dimensional feature vectors even for small values of *n*, which makes the system difficult to implement with conventional memory resources. To overcome this problem, the size of amino acid alphabet is gradually reduced for increasing values of *n* such that the resulting vector for each *n*-peptide composition will have a dimension lower than a constant value of *t*, so getting an upper bound on the space complexity by $O(kt)$. In other words, we use an alphabet size of *r* that satisfies the condition $r^n < t$ for *n*-peptide composition. Not only providing an efficient space complexity, this scheme also allows the evaluation of possible mismatches in longer *n*-peptides, which is a natural case in the evolution of proteins. Table 3 gives the reduced amino acid alphabet sizes and resulting feature vector dimensions for different values of *t* to be used in the construction of *n*-peptide compositions.

Table 3. The amino acid alphabet sizes and resulting feature vector dimensions used for each *n*-peptide composition with varying thresholds of dimension

n	*t*=1000		*t*=5000		*t*=10000	
	Alphabet Size	Vector dimension	Alphabet size	Vector dimension	Alphabet size	Vector dimension
1	20	20	20	20	20	20
2	20	400	20	400	20	400
3	10	1000	15	3375	20	8000
4	5	625	8	4096	10	10000
5	3	243	5	3125	6	7776
6	3	729	4	4096	4	4096

Another problem with the use of *n*-peptide compositions is the exponential time complexity of the protein vectorization. A naive algorithm that searches all possible *n*-peptides in a protein sequence of length *m* has a time complexity of $O(m20^n)$. We use a hash structure indexed by a sorted array of all possible *n*-peptides and sequentially traced the sequence to update the counts of the observed *n*-peptide. With this scheme, the time complexity is reduced to $O(mn)$. Since we use only small values of *n*, the time complexity can be simply regarded as $O(m)$.

We use the amino acid grouping method provided by Murphy, Wallqvist and Levy (2000) in order to reduce the alphabets. These alphabets have been produced using statistical techniques based on the information of certain BLOSUM matrices (Henikoff and Henikoff, 1992) and justified by well-known biochemical amino acid classes. The procedure for grouping similar amino acids has two steps: first, the correlation coefficients between substitution matrix elements are calculated for all pairs of amino acids by the following formula, where $M(x,y)$ denotes the corresponding BLOSUM matrix entry for two amino acids *x* and *y* (BLOSUM matrices are described in detail in Section 2.2.1),

$$C(a,b) = \frac{\sum_{i=1}^{20} M(a,i) \cdot M(b,i)}{(\sum_{i=1}^{20} M(a,i) \cdot M(a,i)) \cdot (\sum_{i=1}^{20} M(b,i) \cdot M(b,i))} \qquad (2.1)$$

In the second step, the two amino acids with the highest correlation coefficient are grouped together, then, the pair with the next correlation is either added to the first group if one member is already on the group or separated into a new group if not. The same process is repeated until all amino acids are partitioned into desired number of groups. Table 4 lists some of the reduced amino acid alphabets obtained from this grouping scheme.

Table 4. Reduced amino acid alphabets used in our method.

Size	Alphabet
20	L V I M C A G S T P F Y W E D N Q K R H
15	(LVIM) C A G S T P (FY) W E D N Q (KR) H
8	(LVIMC) (AG) (ST) P (FYW) (EDNQ) (KR) H
6	(LVIM) (AGST) (PHC) (FYW) (EDNQ) (KR)
5	(LVIMC) (AGSTP) (FYW) (EDNQ) (KRH)
4	(LVIMC) (AGSTP) (FYW) (EDNQKRH)
3	(LVIMCAGSTP) (FYW) (EDNQKRH)
2	(LVIMCAGSTPFYW) (EDNQKRH)

2.2 Empirical Representation by Pairwise Similarities

In 2003, Liao and Noble suggested a new but quite simple methodology to represent proteins by a fixed-length vector. In their scheme, each protein is represented by a set of pairwise similarity scores between the target protein and some other annotated proteins. This scheme does not consider any observed property of amino acid sequences but rather uses an empirical feature map on the basis of a set of known proteins. Each protein, P_x, in the data set is vectorized by $\varphi(P_x)$ with the following equation:

$$\varphi(P_x) = [S(P_x,P_1),\ S(P_x,P_2),\ldots,S(P_x,P_n)] \qquad (2.2)$$

where P_i is i^{th} protein sequence in the labeled protein set, n is the total number of proteins in the labeled set, and $S(P_x,P_i)$ is the alignment score between any protein sequences P_x and P_i. Liao and Noble have shown in their work that the sequence alignment with dynamic programming is the most accurate way of inferring similarity between two sequences as a result of remote homology detection tests. This result was not surprising because dynamic-programming-based alignment had been previously shown to be the optimal solution for sequence similarity in subject to a given objective function. However, as Liao and Noble also stated in their article, the system works very slowly due to the computational inefficiency of dynamic programming algorithm. It is likely for this reason that this method could not find any application in later studies in protein classification.

In this study, two efficient methods for inferring similarity between protein sequences are proposed and they are adopted in the empirical feature representation scheme introduced by Liao and Noble. Main goal of these attempts is to make this powerful strategy practical to use in real-world applications. After giving the fundamental concepts of sequence alignment, following two sections describe the novel methods for measuring sequence similarity.

2.2.1 Sequence Alignment

The degree of similarity of two protein sequences is determined by their alignments. The problem of sequence alignment has been studied extensively in previous three decades (Needleman and Wunch, 1970; Hirschberg, 1977; Smith and Waterman, 1981; Myers and Miller, 1988; Myers,1991; Lipman and Pearson, 1985; Altschul, Gish, Miller, Myers and Lipman, 1990; Delcher, Kasif, Fleishman, Peterson, White and Salzberg, 1999).

2.2.1.2 Pairwise Alignment

The pairwise sequence alignment is the problem of comparing two sequences while allowing certain mismatches between the two.

Definition Given the sequences $A=a_1a_2...a_m$ and $B=b_1b_2...b_n$, the alignment of *A* and *B* finds a set of evolutionary operations, called as *mutations*, that converts *A* to *B* and minimizes the sum of the cost of the total operations, which, in fact, maximizes the *similarity*.

The operations may be *substitution*, replacing one letter to another; *deletion*, removing one letter; or *insertion*, adding a new letter. Since insertion and deletion are dual of each other, the term *indel* is commonly used.

```
AC--BCDDDB
 |  | | |
-CADB-DAD-
```

Figure 3. An example alignment

Informally, an alignment of two strings *A* and *B* is obtained by first inserting chosen spaces, either into or at the ends of *A* and *B* so the length of the strings will match, and then placing the two resulting strings one above the other so that every character or space in one of the strings is matched to a unique character or a unique space in the other string. In Figure 3, one possible alignment of the sequences "ACBCDDDB" and "CADBDAD" is given, where (-) denotes a gap and (|) shows a match of two symbols.

There are two variants of biologically motivated sequence alignment; global alignment and local alignment. Global alignment optimizes the score for similarity (or distance) over the full length of both sequences. It is appropriate when the two sequences are known to be similar and of roughly the same length. Needleman and Wunsch (1970) are the first who formulized the problem.

Definition Let *A* and *B* are two sequences with $|A|=m$ and $|B|=n$, *m* and *n* are roughly the same value, $\sigma(a_k,b_k)$ is the score of the alignment of character a_k with character b_k, and $V(i,j)$ is the optimal score of the alignment of $a_1a_2...a_i$ and $b_1b_2...b_j$ (0<=*i*<=n, 0<=*j*<=m). Then,

$$V(i,j) = max \begin{cases} V(i-1,j-1)+ \sigma(a_i,b_i) \\ V(i,j+1)+ \sigma(a_i,-) \\ V(i-1,j)+ \sigma(-,b_i) \end{cases} \qquad (2.3)$$

where – refers to a space character.

Representing $V(i,j)$ as an *n x m* matrix, with indices *i* and *j*, the alignment with highest score is found by tracing back through the matrix. The space and time complexity of alignment is $O(mn)$; the algorithms needs $O(mn)$ time and $O(m+n)$ space for filling the matrix and $O(mn)$ space for tracing back the alignment. Hirschberg first discovered a linear-space algorithm for the sequence alignment (Hirschberg, 1977), which was then extended with use of edit graphs by Myers and Miller (Myers and Miller, 1988). However, reducing space requirement increased the time complexity.

Definition Let *A* and *B* are two sequences, the optimal score of the local alignment of $a_1a_2...a_i$ and $b_1b_2...b_j$ (0<=*i*<=n, 0<=*j*<=m) is, denoted by $V(i,j)$, given by the recursive formula;

$$V(i,j)= max \begin{cases} V(i-1,j-1)+ \sigma(a_i,b_i) \\ V(i,j+1)+ \sigma(a_i,-) \\ V(i-1,j)+ \sigma(-,b_i) \\ 0 \end{cases} \qquad (2.4)$$

Local alignment finds the similar subregions and optimizes the alignment score for them. It is appropriate when it is not known in advance if the sequences being compared are similar. The Smith & Waterman algorithm is the most widely used local alignment algorithm (Smith and Waterman, 1981). The most important trick for local alignment is to rate the expectation value for a random alignment negative. That causes random alignments and other stretches of mismatches to decrease the path score. Best local alignment is identified by storing the entire *V(i,j)* matrix, finding the maximum element, then tracing back through the matrix until the path score has dropped to zero.

Alignment Heuristics

Dynamic programming methods are feasible when the sequences are short. However, for comparing a sequence with a whole database of hundreds of sequences, they are inefficient. To solve this problem some heuristic approaches such as FASTA (Lipman and Pearson, 1985) or BLAST (Altschul et al., 1990) are used. FASTA considers exact matches between short substrings of two sequences. If a significant number of such exact matches is found, FASTA uses the dynamic programming algorithm to compute optimal alignments. BLAST is another heuristic based on a similar idea. It focuses on no-gap alignment of a certain, fixed length *k*. Rather than requiring exact matches, BLAST uses a scoring function to measure similarity. In particular for proteins, one can argue that segment pairs with no gaps and a high similarity score indicates regions of functional similarity. For a given threshold parameter ***t***, BLAST reports to the user all database entries which have a segment pair with the query sequence that scores higher than ***t***.

Score Matrices for Proteins

A simple way to define $\sigma(a_i,b_j)$ scores is to set it to +1 for matches, -1 for mismatches and –2 for indels. However, depending on the context, some changes may be more plausible than others. The exchange of an amino acid by the one with similar properties (size, charge, etc.) may be more possible than the exchange by the one with opposite properties. Several scoring schemes have been proposed to describe different exchange possibilities. Most popular of those schemes is PAM (Point Accepted Mutations) matrices. In the 1970s, a research team lead by M. Dayoff carefully studied the evolution of sequences of amino acids and constructed some matrices to show the exchange probabilities between individual amino acids (Dayoff, Schwartzi and Orcutt, 1978). PAM or PAM1 is the length of time for 1% of the amino acids to mutate. One estimate that a PAM is that about a billion years. PAM1 matrix is a matrix with column and row headings of the amino acids where matrix cells refer to the amount of evolution over one PAM period of time, or for one mutation per hundred amino acids. Some extensions are possible over PAM1 matrix. A PAM70 matrix, for example, contains scoring information on the amount of evolution over 70 PAM period of time. Any of PAM<X> matrix can be obtained by raising the PAM1 matrix to the X. power. Figure 4 shows a smaller part of the PAM70 matrix. As shown in the matrix, exchange of E by D is more possible than exchange of E by C, since E-D score, 3, is higher than E-C score, -9. Choosing a proper PAM matrix depends on the application data. If the sequences are known to be similar at advance, a PAM matrix with smaller index is preferable. Otherwise, a high index should be chosen. BLOSUM series of substitution matrices were later introduced by Henikoff and Henikof (1992) which is believed to be more precise for distantly related protein sequences.

	A	R	N	D	C	Q
A	5	-4	-2	-1	-4	-2
R	-4	8	-3	-6	-5	0
N	-2	-3	6	3	-7	-1
D	-1	-6	3	6	-9	0
C	-4	-5	-7	-9	9	-9
Q	-2	0	-1	0	-9	7

Figure 4. A part of PAM70 matrix.

Multiple Sequence Alignment

Multiple sequence alignment is simply the extension of pairwise global alignment with *k* sequences instead of two.

Definition Given a set $S=\{s_1,s_2,\ldots,s_k\}$ of k sequences, such that the sequences are known to be similar to each other, a multiple alignment M maps the set S to a new set $S'=\{s_1',s_2',\ldots,s_k'\}$, such that;

i. all the sequences in S' are of equal length, and

ii. the removal of spaces from s_i' leaves s_i, for $1 \leq i \leq k$.

Since there are so many possibilities for such an alignment, the next problem is to design a scoring function which determines the quality of the alignment. If we define a distance (similarity) function $\sigma(x,y)$, which measure the distance (similarity) between the individual characters, a pairwise alignment score between s' and t';

$$\sum \sigma(s_i',t_i'),\ 1 \leq i \leq L,\ L=|s'|=|t'|$$

is to be minimized (maximized). $\sigma(x,y)$ score is usually obtained by a substitution matrix, like PAM, described in previous sections.

For the alignment of k sequences, a sum of pairs *(SP)* score of multiple alignment M is defined as the sum the scores of all pairwise alignments induced by M. Then, the multiple alignment is to find the alignment M of $\{s_1,s_2,\ldots,s_k\}$ which has the minimum possible *(SP)* score for these k sequences.

The number of available methods for multiple sequence alignment has steadily increased over the last 20 years. The methods can be grouped into three; progressive algorithms, exact algorithms and iterative algorithms.

The progressive algorithms (Higgins, Thompson, Gibson, 1994, Löytynoja and Milinkovitch, 2003) attempt to optimize the weighted sum of the pairwise similarities. The sequences are added one by one to the multiple alignment according to the order indicated by a precomputed dendogram. Sequence addition is made using a pairwise sequence alignment method, such as dynamic programming. This method is simple and effective, however, it has a problem that once a sequence is added to alignment, it will never be modified even if it conflicts with the sequences added later.

The exact algorithms attempt to find optimal alignment, instead of approximating to it. The use of dynamic programming in *k* dimensional space (where *k* stands for the number of sequences) is very inefficient in time and space. This is acceptable for a maximum of three sequences. This limit can be extended a little bit further by finding a way to identify in advance the portion of the hyperspace that does not contribute to the solution and exclude it from computation (Carillo and Lipman, 1988).

The iterative algorithms, in general, are based on the idea that the solution to a given problem can be computed by modifying an already existing sub-optimal solution. The modifications can be made using dynamic programming or in a stochastic way. There are some proposed methods which use Hidden Markov Models (Krogh, Brown, Mian, Sjölander and Haussler, 1994), simulated annealing (Kim, Pramanik and Chung, 1994), Gibbs sampling (Lawrence, Altschul, Boguski, Liu, Neuwald and Wootton, 1993) or genetic algorithms (Notredame and Higgins., 1996). Among these, HMM approach is limited in practice to cases with many sequences (>100). The others give effective results with properly chosen parameters.

2.2.2 A Conservative Definition for Similarity: Maximal Unique Matches

Definition Given two sequences *A* and *B*, a substring that appears only once in both *A* and *B* is called a *unique match* between *A* and *B*, and a unique match is called as a *maximal unique match* if it is not contained in any longer unique match.

With this definition, a maximal unique match (MUM) can be considered as one of the core part in an alignment and yields an important evidence for the homology between the sequences. The definition is more strict than the alignment since it does not allow any substitution and repetition in the sequences when evaluating the similarity between them. However, when the we deal with the sequences that have less similarity, the allowance of any mutation may lead to by-chance matches between the sequences. Therefore, this strict definition is expected to show more evidently the local relationships between the divergent proteins. The definition of MUM has been originally introduced by Delcher et al. (1999) to accelerate the alignment of long DNA strands. In this study, we adopt their definition for the protein sequences to represent the conservative relationships between them.

Once identified, the length of all non-overlapping maximal unique matches can be used to compare two protein sequences; $S(P_x,P_i)$ score in Equation 2.1 is replaced by the total length of all MUMs, $M(P_x,P_i)$, for protein vectorization;

$$\varphi(P_x) = M(P_x,P_1),\ \ M(P_x,P_2), \ldots, M(P_x,P_n) \qquad (2.5)$$

The simple sum of all MUM lengths leads to a problem when two or more MUMs overlap since the overlapping residues would be counted more than once. To overcome this problem, we modify the definition of $M(P_x,Pi)$ as the number of the residues contained in a maximal unique match between P_x and P_i.

Finding MUMs

To find the maximal unique matches, a special data structure called suffix tree is used. A suffix tree is a compact tree that stores all suffixes of a given text string. It is a powerful and versatile data structure which finds application in many string processing algorithms (Gusfield, 1997). An example suffix tree for "*abab$*" is shown in Figure 5.

Definition Let A be string of n characters, $A=s_1s_2...s_n$, from an ordered alphabet Σ except s_n. Let $\$$ be a special character, matching no character in Σ, and s_n be $\$$. The suffix tree T of A is a tree with n leaves such that;

- Each path from the root to a leaf of *T* represents a different suffix of *A*.
- Each edge of *T* represents a non-empty string of *A*.
- Each non-leaf node of *T*, except the root, must have at least two children.
- Substrings represented by two sibling edges must begin with different characters.

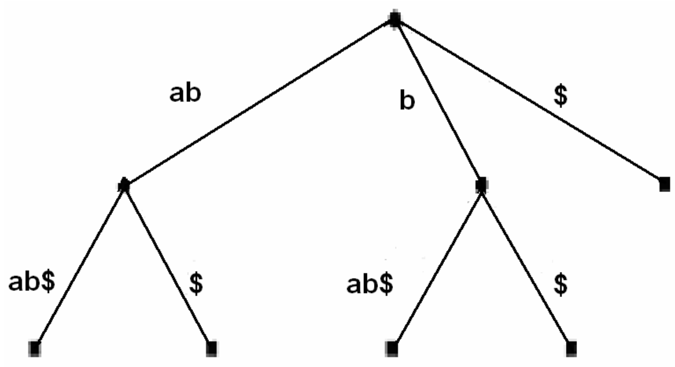


Figure 5. Suffix tree of "abab$".

The definition of a suffix tree can be easily extended to represent the suffixes of a set *{A_1,A_2,...,A_n}* of strings. This kind of suffix tree is called as a generalized suffix tree.

To find maximal unique matches between any two sequences, first, a generalized suffix tree (Bieganski, Riedl, Carlis and Retzael, 1994) is constructed for the sequences. This is simply done by concatenating two sequences with a dummy character (not contained in the alphabet) between them and constructing a suffix tree for the newly created sequence. In our representation, a maximal unique match is a *maximal pair* in the concatenated sequence one of which appears before the dummy character and the other appears after that. The algorithm to find maximal pairs is given by Gusfield (1997). We used a variation of this algorithm considering the fact that each of the pair should appear in different sequences. The leaves of the suffix tree are numbered according to the position of the suffix which they represent. The algorithm used to find the unique matches is given in Figure 6.

```
# finding unique matches (S_unique)
s1 <- first sequence
s2 <- second sequence
sc <- concatenate s1, #, s2 (# is a dummy character)
ST <- construct suffix tree for sc
sep <- leaf number of the node representing #s2
for each node, n, in ST
        if n has exactly two leaf childs (c1,c2) then
                L1 <- leaf number of c1
                L2 <- leaf number of c2
                if ((c1>sep and c2<sep) or (c1<sep and c2>sep)) then
                        n is a matching node
                        add the path from root to n to S_unique list
                end if
        end if
end for
```

Figure 6. Algorithm for finding unique matches

Given a generalized suffix tree for two sequences, unique matches between them can be found linear time. Since the construction of a suffix tree can also be completed in linear time (Weiner, 1977; McCreight, 1976; Ukkonen, 1995), the algorithm for finding unique matches would have a linear time complexity. The maximality of unique matches can be determined simply by mismatches at their left and right ends.

2.2.3 Pairwise Probabilistic Suffix Trees (PSTs)

The PST method was introduced by Bejerano and Yona (2000) to model the protein families. The original PST model was based on identifying significant short segments among the many input sequences, regardless of the relative positions of these segments within the different proteins. The model induces a probability distribution on the next symbol to appear right after a length segment with a length of no more than a predefined value, say *L*. To classify a sequence into one of the families, a separate PST is constructed for each family in the data set, and according to the probability distribution over PST, a probability that the sequence belongs to that family is assigned to the query sequence. By comparing this probability score against a threshold value (predetermined by applying an equivalence criterion), the sequence is determined as belonging to that family or not.

Defined over a finite alphabet, a PST contains edges labeled by a single symbol of the alphabet, such that no symbol is represented by more than one edge branching out of any single node. Each node in the PST, varying in degree between zero and the size of the alphabet, is labeled by a string, which is the one generated by traversing the tree from that node to the root. The nodes are also attached with a conditional empirical probability vector of the same size as the alphabet. The probability distribution vector contains the probability of each symbol to appear after a subsequence represented in the current node.

An example PST is given in Figure 7. For example, the probability vector associated with the node "*bea*" is (0.1, 0.1, 0.35, 0.35, 0.1). In other words, according to given PST, the probability of observing the symbol *a* after the segment *bea* is 0.1, whereas the probability of observing *d* after *bea* is 0.35.

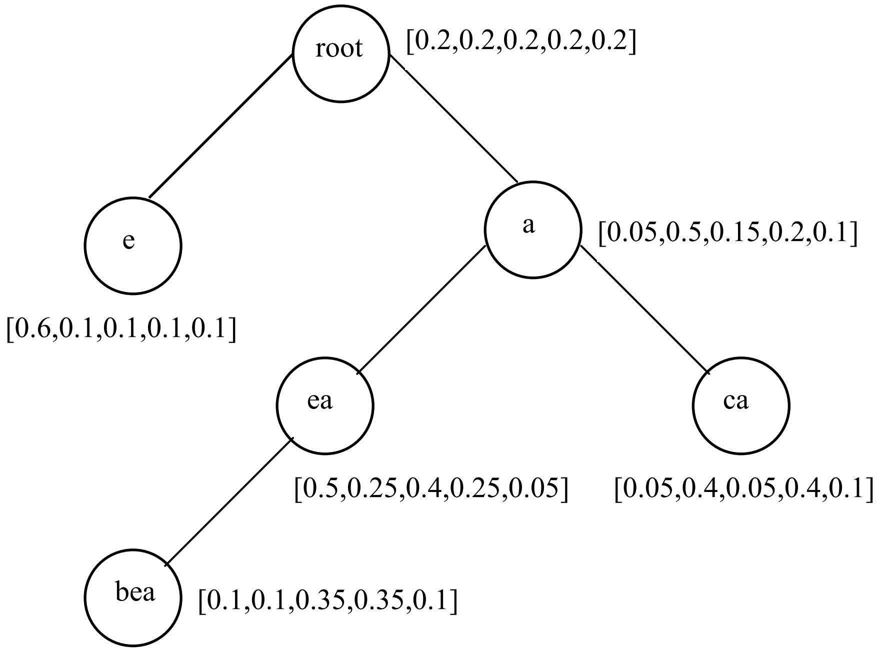


Figure 7. An example PST over the alphabet {a,b,c,d,e}

In this study, we propose a variant form of the PST method for pairwise sequence comparison. To infer the similarity between two sequences, the query sequence is predicted using the PST constructed from the source sequence and the predicted probability score is taken as the similarity score between them. The probability of a sequence is calculated by the average of the probability of each letters:

$$\wp(P_x,P^T_y)= (\gamma_{s1}(x_1)+ \gamma_{s2}(x_2)+\ldots+ \gamma_{sk}(x_k)) / k \tag{2.6}$$

where $\wp(P_x,P^T_y)$ is the probability score of protein sequence P_x over the PST of protein sequence P_y, $\gamma_i(x_i)$ is the probability of i^{th} symbol x_i in protein sequence P_x, calculated by scanning the tree to search for the longest subsequence (s_i) that appears in the tree and ends just before the symbol x_i, and k is the length of the protein sequence P_x.

To adopt the new sequence similarity measure into the feature representation scheme introduced by Liao and Noble, we built a separate PST for each sequence in the training set and modified the protein vectorization given in Equation 2.2 by the following:

$$\varphi(P_x) = [\wp(P_x,P^T_1),\ \wp(P_x,P^T_2),\ldots,\ \wp(P_x,P^T_n)] \tag{2.7}$$

An ideal PST for a single sequence contains a leaf labeled by whole sequence and a node for each prefix of the sequence. Since this leads to an enormous amount of memory use even for small sequence input, a PST construction parameter, so-called *short memory length*, has been provided in Bejerano and Yona's implementation. This parameter, denoted by L, restricts the degree of the PST nodes by a constant value, which also refers to the maximum length used for the suffixes to be used in the prediction step. Another parameter in the PST construction algorithm is P_{min}, which denotes the minimum probability with which strings are required to appear in the PST input to be represented by a node in the resulting PST. This probability is calculated by dividing the number of appearances of the related string by the number of all possible strings with the same length. Since the appearance of a string only once in the source sequence is even essential for pairwise comparison, we set P_{min} to zero in our pairwise application. Although many other parameters have been provided by Bejerano and Yona to optimize the space complexity of the PST construction algorithm with multiple sequence input, these did not require further optimization since we use only a single-sequence input for pairwise comparison. Thus, we adjust all these parameters so that all strings shorter than L are represented by a node in the resulting PST.

Modifications on PST

Some modifications in the prediction phase of the PSTs are proposed for single-sequence application of PSTs into pairwise comparison.

1. Maximizing probability: One of the problems with PSTs is that the probability of observing a specific residue after a significant segment with a length of L may be zero. Since the original PST approach is family based, this scoring may be reasonable to penalize a residue that diverges too much from the family profile. However, in our pairwise application, penalizing a residue that has a possibility of getting a higher score with a shorter suffix is not reasonable. To overcome this, we search the maximum possible probability score for each residue in the proteins. This is done by comparing the probabilities obtained for all suffixes preceding the target residue.

2. Average degree as a multiplication factor: Considering that the use of larger suffixes preceding the residue under consideration may constitute a better indication for homology, we put the average degree of the nodes used in the prediction of each residue within the sequence as a multiplication factor to the final similarity score:

$$\wp(P_x, P^T{}_y) = ((\gamma_{s1}(x_1) + \gamma_{s2}(x_2) + \ldots + \gamma_{sk}(x_k)) / k) * (\sum \omega_i)/k \qquad (2.8)$$

where ω_i as the length of the preceding segment used for calculating the probability of next symbol in position *i*. This scheme rewards the appearance of longer segments in the query sequence.

3. Two-way similarity: Until now, all the similarity calculations are defined in one way. That is, to compare the protein sequence P_x with the protein sequence P_y, we have used the PST constructed from the protein sequence P_y to find the probability of protein sequence P_x to be produced from that PST. Owing to the nature of PSTs, it is obvious that the probability of protein sequence P_y to be produced from the PST of the protein sequence P_x will be different. This problem becomes more serious when the length difference between the sequences is high. Considering this, we use both scores in our vectorization:

$$\varphi(P_x) = [(\wp(P_x, P^T{}_1), \wp(P_1, P^T{}_x)), (\wp(P_x, P^T{}_2), \wp(P_2, P^T{}_x)), \ldots, (\wp(P_x, P^T{}_n), \wp(P_n, P^T{}_x))] \qquad (2.9)$$

4. Adding biological considerations. The PST based scoring scheme does not consider any biological priori information, instead uses only the probability distribution on the source and query sequences. The biological considerations can also be incorporated to the PST scoring scheme using well-known substitution matrices such as PAM or BLOSUM (see Section 2.2.1). This approach aims to enhance the similarity score by allowing the possible point mutations. Assuming that any mutation between two amino acid *a* and *b* has a substitution score of $\partial(a,b)$, we can replace the probability of the symbol $\gamma_s(a)$ after a segment *s* with the $\gamma_s(b)$ if $\gamma_s(b).\ \partial(a,b) > \ \gamma_s(a).\ \partial(a,a)$. In this case, the probability of a sequence is calculated by

$$\wp(P_x, P^T{}_y) = (\gamma'_s(x_1) + \gamma'_s(x_2) + \ldots + \gamma'_s(x_k)) \,/\, k \qquad (2.10)$$

where

$$\gamma'_s(x_i) = maxof\,\{\ \gamma_s(x_i),\ (\gamma_s(aa_1).\partial(xi,aa_1)/\partial(x_i,x_i)), \ldots,\ (\gamma_s(aa_{20}).\partial(x_i,aa_{20})/\ \partial(x_i,x_i))\ \}$$

and aa_k is the k^{th} amino acid in the protein alphabet. It should be noted that this scheme can be applied only when all substitution scores between amino acids are positive. Hence, a substitution matrix whose values are normalized between 0 and a positive number should be used if this modification is intended to apply. Another remark is that this modification can not be used at the same time with the first modification which is suggested to get the highest possible score for each residue. Thus, the choice between two modifications is left as an option in the PST prediction step.

The protein sequence representation scheme based on the PST-based sequence similarity scores is utilized in SVM-based remote homology detection and subcellular localization prediction systems. The similarity scores for N-terminal parts of the sequences are also employed in subcellular localization predictions knowing that the localization process is oriented by some targeting signals generally occurring in this region (Emanuelsson, Nielsen, Brunak and Gunnar, 2000). The accuracy and efficiency of this approach are evaluated in the following chapters.

CHAPTER 3

Remote Homology Detection

Two proteins are said to be homolog if they share a common evolutionary origin. Homology information is important in that it may imply a common structure and function between two proteins. Since we are often supplied only the sequence information, inferring homology using solely the sequence has been one of the central problems in computational biology. If we are able to find some number of similar proteins whose structural and functional analyses are already completed, the target protein can be easily annotated using the assumption that the sequence is the main determinant of the structure. However, there are two main problems with this argument. First, the target protein may be entirely new and its structure is different from all of the proteins available in the databases. Second, in spite of the weak similarity between two protein sequences they may still have evolutionary relationships. The first problem is a bottleneck of computational biology and there is no method that works well at the moment. The second problem is known as *remote homology detection* problem, and various methods have been proposed in recent years. In spite of several successful attempts, they are either computationally inefficient or insufficient to work for all cases. The previous methods can be grouped into three stages; pairwise methods, generative methods and discriminative methods. A taxonomy of the methods are given in Figure 8 and the following section is devoted to their descriptions.

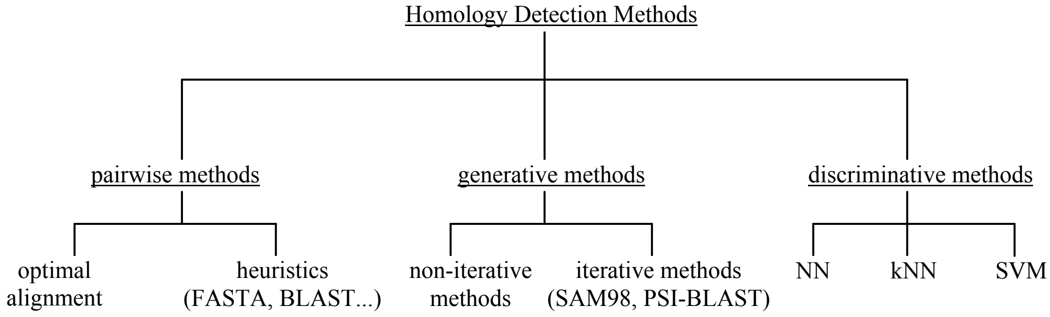


Figure 8. A taxonomy of the existing homology detection methods

3.1 Previous Studies

The early methods for homology detection were based on the pairwise sequence similarity inferred by dynamic programming based sequence alignment (Smith and Waterman, 1981). While the dynamic programming method finds an optimal score for similarity according to a predefined objective function, it suffers from long computation times for relatively long sequences. To speed up the alignment, some heuristic methods, such as BLAST (Altschul et al. 1990), have been developed to find a near-optimal alignment within a reasonable time. The general assumption is that two proteins are homolog if the sequence identity (the percentage of identical residues after the alignment of two sequences) is above 40%. The problem with pairwise sequence alignment is biological inaccuracy of evaluation with respect to only one known protein homolog. Considering only one protein to annotate a newly sequenced protein may lead to biologically uncertain results. This uncertainty comes from the fact that the *twilight zone* of sequence alignment sets a boundary for confidence levels for the detection of evolutionary relatedness of proteins (Rost, 1999). In most alignments this twilight zone falls between 20-40% sequence identity. Despite two proteins are not similar in terms of their sequences, i.e. the sequence identity is below 40%, they may still share some important structural or functional features, which actually refers to remote or distant homology between proteins.

To take apart from the twilight zone of pairwise sequence alignment, family based comparisons have been proposed (Grundy, 1998). To improve the sensitivity of homology results, a representative set of sequences from the family is incorporated into the comparison, that is, the new protein is aligned concurrently with all (or some) of the protein sequences in a specified family. The utilization of multiple comparisons have increased three times the sensitivity of homology detection compared to pairwise comparisons (Park et al., 1998). Indeed, most of the proteins are classified within a protein family in available databases and the actual annotations are made over those families, thus, it seems more appropriate to use all family knowledge in homology modeling.

Since the concurrent alignment of new sequence to all members of a family may disrupt the actual properties of the family, an alternative approach would be to construct family *profiles* and align the new sequence with this profile (Gribskov, Lüthy and Eisenberg, 1990, Eddy, 1998). For this, the common properties of the sequences in a family are encompassed in a family profile and the new sequence is tested to check whether or not it belongs to the family by comparing to its profile. These methods are based on the similarity statistics derived upon more than one homolog examples, that is, all statistical information is generated from a set of sequences that are known or posited to be evolutionary related to another. These probabilistic methods are often called as *generative* because they induce a probability distribution over the protein family and try to generate the unknown protein as a new member of the family from this stochastic model.

Additional accuracy can be gained by iteratively searching the available database for homologs and refining the central profile model in each iteration. SAM-T98 method is an example of iterative family refinement methods (Karplus, Barett and Hughey, 1998). In this method, sequences are first aligned to an initial model which is constructed based on some background distributions and then the model is improved iteratively by aligning the sequences to the current model to which the new statistical results are incorporated. Another iterative method PSI-BLAST deploys BLAST search and refine the results iteratively (Altschul et al., 1997).

Integrated sequence/structure alignments are iteratively performed for the search of homologs in another method (Walqvist et al., 2000).

The main problem with generative approaches is the fact that they produce so much false positives, that is, they report a number of homologs though they are not. For this reason, the recent works on remote homology detection have begun to use a discriminative framework to make separation between homolog (positive) and non-homolog (negative) classes. In contrast to generative methods, the discriminative methods focus on learning the combination of features that discriminate between the classes. These methods attempt to establish a model that differentiates between positive and negative examples. In other words, non-homologs are also taken into account.

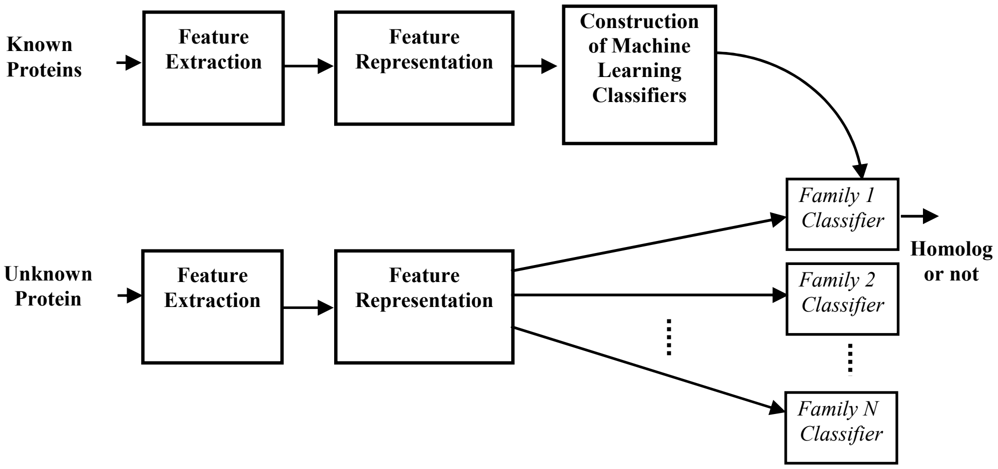


Figure 9. Discriminative homology detection model

In discriminative homology detection methods, there are two main phases: training and testing. The training phase constructs a machine learning classifier for the specified family, and the testing phase uses this classifier to decide whether the test protein is belonging to this family or not. In general, a machine learning classifier is constructed for each family in the database and the protein is checked if it is belonging to any of those known protein classes. Both phases require the extraction of some informative features from the protein sequence and the representation of these features in a suitable way. Figure 9 gives an overview of the discriminative homology detection approach.

The current methods using the discriminative approach differ in the feature extraction methods, the feature representation schemes and the type of the machine learning classifiers they have used. Among k-nearest neighbor method, neural networks and support vector machine, the last one has been reported as outperforming the others in many applications concerning with protein classification (Liao and Noble, 2003; Ding and Dubchak, 2001).

Discriminative methods are more successful than generative methods in terms of separation accuracy between true positives (homologs that are correctly predicted) and false positives (non-homologs that are incorrectly predicted as homolog). However, the training and testing phases require so much time with conventional workstations, which makes them inappropriate to use in practice. Thus, more efficient methods are required that preserve the classification accuracy.

First discriminative approach (SVM-Fisher) represents each protein by a vector of Fisher scores extracted from a profile Hidden Markov model constructed for a protein family and utilizes SVMs to classify the protein with those feature vectors (Jaakola, Diekhans and Haussler, 2000). A recent and more successful work, called SVM-Pairwise (Liao and Noble, 2003), combines the sequence similarity with the SVMs to discriminate between positive and negative examples. Detailed description of the feature representation based on pairwise similarity scores is given in Section 2.2.3. In SVM-Pairwise, both the training and test sets include positive and negative examples. This method was tested for dynamic-programming-based alignment scores and BLAST scores. Note that the latter one is referred as SVM-BLAST in the following sections. SVM-Pairwise approach is among the best methods in terms of accuracy, but it suffers from computational inefficiency since the alignment takes too much time for long sequences. Another drawback of this approach is that the alignment may force some residues to match even if they are evolutionary not related.

3.2 Systems and Methods

In this study, three new methods for feature vectorization; (1) n-peptide compositions, (2) MUM-based similarity scores, and (3) PST-based similarity scores, described in Chapter 2 are employed for remote homology detection in the discriminative framework explained above. For convenience, the new systems are named as SVM-*n*-peptide, SVM-MUM and SVM-PST respectively. A comparative analysis with the other methods is presented in the following sections.

3.2.1 Binary Classification with SVMs

To discriminate between positive and negative examples SVMs are used. SVMs are binary classifiers that work based on the structural risk minimization principle (Vapnik, 1995; Vapnik and Cortes, 1995). They have been extensively used and shown to be a powerful tool in many bioinformatics problems (Hua and Sun, 2001; Ward, McGuffin, Buxton and Jones, 2003; Yuan, Burrage and Mattick, 2002; Karchin, Karplus and Haussler, 2002; Bock and Gough, 2001). An SVM classifier is generated by a two-step procedure: first, the high-dimensional input space of the SVM is non-linearly mapped into a higher dimensional feature space. In the second step, a linear hyperplane is constructed in this feature space with the largest possible margin separating the classes of the data. The points classified by the SVM can be divided into two groups; support vectors and nonsupport vectors. Nonsupport vectors are perfectly classified by the hyperplane and are located outside the separating margin. Parameters of the hyperplane do not depend on them, even if their position is changed, provided that these points will stay outside the margin. Support vectors are the points that are difficult to classify and therefore they are used to determine the exact position of the hyperplane. In other words, support vectors contain the information for the classification task. The main advantage of SVM is its better generalization ability owing to fact that it finds the separating hyperplane with the largest margin using support vectors, as opposed to neural networks at which all possible hyperplanes are evaluated. Thus, SVM is said to be less prone to overfitting than other classifiers.

To train the SVMs, an open-source software called SVM-Gist, available at www.cs.columbia.edu/compbio/svm, is used. In the SVM-Gist software, a kernel function acts as the similarity score between pairs of input vectors. The base kernel is normalized in order to make that each vector has a length of 1 in the feature space, that is,

$$K(X,Y) = \frac{X.Y}{\sqrt{(X.X)(Y.Y)}} \tag{3.1}$$

where X and Y are the input vectors, K(.,.) is the kernel function, and "." denotes the dot product. This kernel is then transformed into a radial basis kernel K'(X,Y), as follows:

$$K'(X,Y) = e^{-\frac{K(X,X)-2K(X,Y)+K(Y,Y)}{2\sigma^2}} + 1 \tag{3.2}$$

where the width σ is the median Euclidean distance from any positive training example to the nearest negative example. Since the separating hyperplane of SVM is required to pass from the origin, the constant 1 is added to the kernel so that the data goes away from the origin. An asymmetric soft margin is implemented by adding to the diagonal of the kernel matrix a value 0.02*ρ, where ρ is the fraction of training set proteins that have the same label with the current protein, as done in the previous SVM classification methods (SVM-Pairwise, SVM-BLAST, SVM-Fisher). The SVM output is a list of discriminant scores corresponding to each protein in the test set.

3.2.2 Implementations and Experimental Setup

n-peptide compositions were constructed with a computer program implemented by standard library functions of Perl language for SVM-*n*-peptide system. To find

MUM-based similarities, another Perl program was written based on the suffix tree algorithm described in Section 2.2.2. To implement SVM-PST, the PST source codes provided by Bejerano (2004) were recompiled with the modifications defined in Section 2.2.3. All programs were run on a Linux system under the supervision of *bash* process.

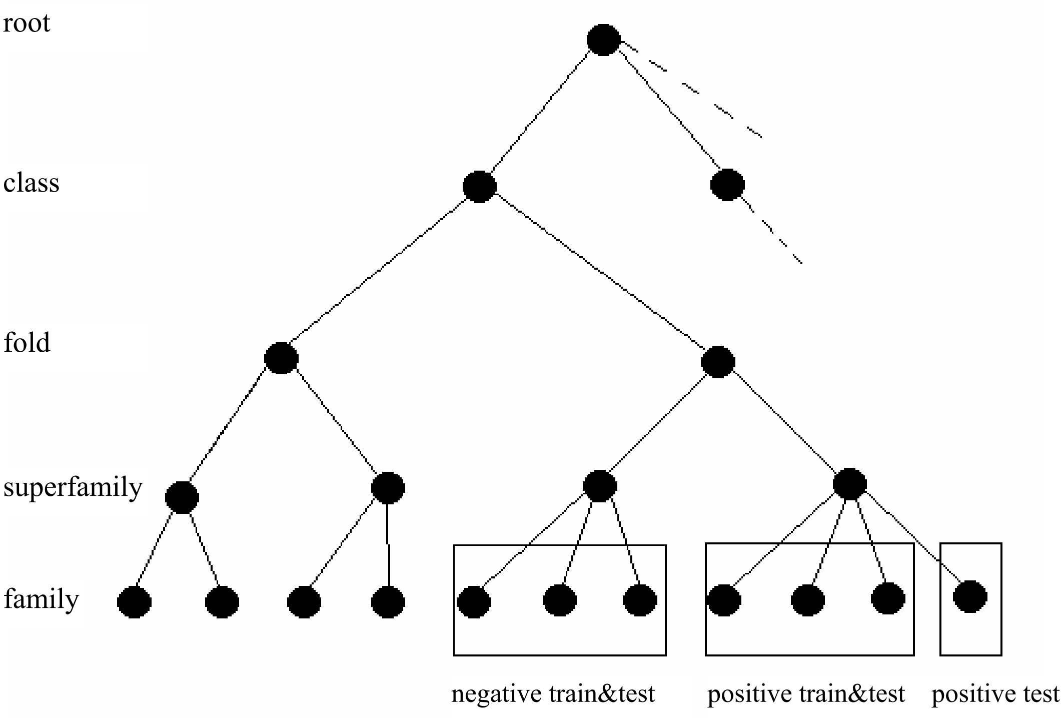


Figure 10. An illustration of how remote homology detection is simulated with SCOP database hierarchy.

The methods were tested to discern their ability to classify proteins into families on a subset of the SCOP family database (Murzin et al., 1995). Remote homology is simulated by excluding same family members from the training set and leaving proteins from same superfamily in the positive set. To make a fair comparison, we worked on the same experimental setup as the one used by SVM-Pairwise and SVM-BLAST. The experiments are performed on a subset of the SCOP1.53 database including no protein pair with a pairwise similarity higher than an E-value of 10^{-25}. The training and test sets were separated as done in Liao and Noble's works resulting with 54 families to test.

Table 5. Number of samples in the SCOP data set.

Family	Positive Set		Negative Set		Family	Positive Set		Negative Set	
ID	Train	Test	Train	Test	ID	Train	Test	Train	Test
1.27.1.1	12	6	2890	1444	2.9.1.4	21	10	2928	1393
1.27.1.2	10	8	2408	1926	3.1.8.1	19	8	3002	1263
1.36.1.2	29	7	3477	839	3.1.8.3	17	10	2686	1579
1.36.1.5	10	26	1199	3117	3.2.1.2	37	16	3002	1297
1.4.1.1	26	23	2256	1994	3.2.1.3	44	9	3569	730
1.4.1.2	41	8	3557	693	3.2.1.4	46	7	3732	567
1.4.1.3	40	9	3470	780	3.2.1.5	46	7	3732	567
1.41.1.2	36	6	3692	615	3.2.1.6	48	5	3894	405
1.41.1.5	17	25	1744	2563	3.2.1.7	48	5	3894	405
1.45.1.2	33	6	3650	663	3.3.1.2	22	7	3280	1043
2.1.1.1	90	31	3102	1068	3.3.1.5	13	16	1938	2385
2.1.1.2	99	22	3412	758	3.32.1.1	42	9	3542	759
2.1.1.3	113	8	3895	275	3.32.1.11	46	5	3880	421
2.1.1.4	88	33	3033	1137	3.32.1.13	43	8	3627	674
2.1.1.5	94	27	3240	930	3.32.1.8	40	11	3374	927
2.28.1.1	18	44	1246	3044	3.42.1.1	29	10	3208	1105
2.28.1.3	56	6	3875	415	3.42.1.5	26	13	2876	1437
2.38.4.1	30	5	3682	613	3.42.1.8	34	5	3761	552
2.38.4.3	24	11	2946	1349	7.3.10.1	11	95	423	3653
2.38.4.5	26	9	3191	1104	7.3.5.2	12	9	2330	1746
2.44.1.2	11	140	307	3894	7.3.6.1	33	9	3203	873
2.5.1.1	13	11	2345	1983	7.3.6.2	16	26	1553	2523
2.5.1.3	14	10	2525	1803	7.3.6.4	37	5	3591	485
2.52.1.2	12	5	3060	1275	7.39.1.2	20	7	3204	1121
2.56.1.2	11	8	2509	1824	7.39.1.3	13	14	2083	2242
2.9.1.2	17	14	2370	1951	7.41.5.1	10	9	2241	2016
2.9.1.3	26	5	3625	696	7.41.5.2	10	9	2241	2016

For each family, the proteins within the family are taken as positive test examples, and the proteins outside the family but within the same superfamily are considered positive training examples. Negative examples are selected from outside of the superfamily. Figure 10 gives the hierarchy SCOP database and an illustration of how the data set was constructed. For each family, at least 10 positive training examples are selected. The negative examples are randomly separated into training and test sets in the same ratio as the positive examples. The numbers of training and test samples are given in Table 5. Corresponding family names are given in Table 6.

3.3 Results and Discussion

Upon completion of SCOP family classification tests, ROC scores were calculated for each family in the dataset with all methods mentioned in this chapter. In Table 7, the ROC scores achieved by the SVM methods based on pairwise similarity scores are given. According to the table, the average ROC score achieved by SVM-PST is higher than SVM-BLAST and SVM-MUM, and it is very close to SVM-Pairwise. When the standard deviation in the ROC scores and the worst case scores are evaluated, SVM-PST is the best among four methods compared. This indicates the superiority of this method to the others in terms of the robustness to the errors due to varying family characteristics. SVM-MUM performs better than SVM-BLAST for most of the families, although MUM-based comparison has a simpler but more conservative definition in comparison with BLAST.

For further investigation of the results, the methods were compared by their relative performances using the plots of the number of families for which a given method exceeds a threshold ROC score. The plots are depicted in Figure 11.

In each plot, a higher curve corresponds to more accurate homology detection performance. According to the curves, SVM-PST performs nearly as well as SVM-Pairwise and better than the others.

Table 6. Family names used in the experiments

ID	Family name
1.27.1.1	Long-chain cytokines
1.27.1.2	Short-chain cytokines
1.36.1.2	Phage repressors
1.36.1.5	Bacterial repressors
1.4.1.1	Homeodomain
1.4.1.2	Recombinase DNA-binding domain
1.4.1.3	Myb
1.41.1.2	S100 proteins
1.41.1.5	Calmodulin-like
1.45.1.2	Bacterial repressors
2.1.1.1	V set domains (antibody variable domain-like)
2.1.1.2	C1 set domains (antibody constant domain-like)
2.1.1.3	C2 set domains
2.1.1.4	I set domains
2.1.1.5	E set domains
2.28.1.1	Legume lectins
2.28.1.3	Galectin (animal S-lectin)
2.38.4.1	Anticodon-binding domain
2.38.4.3	Single strand DNA-binding domain, SSB
2.38.4.5	Cold shock DNA-binding domain-like
2.44.1.2	Eukaryotic proteases
2.5.1.1	Plastocyanin/azurin-like
2.5.1.3	Multidomain cupredoxins
2.52.1.2	Phosphotyrosine-binding domain (PTB)
2.56.1.2	Fatty acid binding protein-like
2.9.1.2	Plant virus proteins
2.9.1.3	Insect virus proteins
3.1.8.1	alpha-Amylases, N-terminal domain
3.1.8.3	beta-glycanases
3.2.1.2	Tyrosine-dependent oxidoreductases
3.2.1.3	Glyceraldehyde-3-phosphate dehydrogenase-like, N-terminal domain
3.2.1.4	Formate/glycerate dehydrogenases, NAD-domain
3.2.1.5	Lactate & malate dehydrogenases, N-terminal domain
3.2.1.6	6-phosphogluconate dehydrogenase-like, N-terminal domain
3.2.1.7	Amino-acid dehydrogenase-like, C-terminal domain
3.3.1.2	FAD-linked reductases, N-terminal domain
3.3.1.5	FAD/NAD-linked reductases, N-terminal and central domains
3.32.1.1	Nucleotide and nucleoside kinases
3.32.1.11	RecA protein-like (ATPase-domain)
3.32.1.13	Extended AAA-ATPase domain
3.32.1.8	G proteins
3.42.1.1	Thioltransferase
3.42.1.5	Glutathione S-transferases, N-terminal domain
3.42.1.8	Glutathione peroxidase-like
7.3.10.1	EGF-type module
7.3.5.2	Spider toxins
7.3.6.1	Long-chain scorpion toxins
7.3.6.2	Short-chain scorpion toxins
7.3.6.4	Plant defensins
7.39.1.2	Nuclear receptor
7.39.1.3	LIM domain
7.41.5.1	Rubredoxin
7.41.5.2	Desulforedoxin

Table 7. Comparison of pairwise-similarity-based feature representations for dicriminative remote homology detection

Family	SVM-Pairwise	SVM-BLAST	SVM-MUM	SVM-PST
1.27.1.1	**0.971**	0.890	0.950	0.948
1.27.1.2	0.918	0.779	0.889	**0.974**
1.36.1.2	0.935	0.870	**0.955**	0.772
1.36.1.5	**0.976**	0.708	0.913	0.957
1.4.1.1	0.968	0.878	**0.980**	0.970
1.4.1.2	0.814	0.810	0.834	**0.898**
1.4.1.3	0.944	**0.999**	0.970	0.842
1.41.1.2	0.999	**1.000**	0.954	0.954
1.41.1.5	**0.998**	0.996	0.927	0.963
1.45.1.2	**0.971**	0.729	0.921	0.732
2.1.1.1	**0.978**	0.949	0.883	0.923
2.1.1.2	**0.994**	0.972	0.970	0.941
2.1.1.3	**0.985**	0.907	0.966	0.991
2.1.1.4	**0.974**	0.947	0.886	0.901
2.1.1.5	**0.832**	0.790	0.799	0.797
2.28.1.1	**0.815**	0.389	0.559	0.612
2.28.1.3	**0.829**	0.412	0.543	0.493
2.38.4.1	0.697	0.702	**0.780**	0.716
2.38.4.3	0.707	0.764	0.681	**0.831**
2.38.4.5	0.877	0.668	0.786	**0.878**
2.44.1.2	0.146	0.925	0.403	**0.935**
2.5.1.1	**0.925**	0.899	0.840	0.744
2.5.1.3	**0.896**	0.826	0.782	0.791
2.52.1.2	0.643	0.641	0.793	**0.829**
2.56.1.2	0.844	**0.878**	0.839	0.838
2.9.1.2	0.874	0.543	**0.887**	0.876
2.9.1.3	0.970	0.909	0.989	**0.996**
2.9.1.4	0.918	0.645	0.926	**0.987**
3.1.8.1	0.963	0.406	**0.990**	**0.990**
3.1.8.3	0.931	0.345	**0.986**	0.976
3.2.1.2	0.838	0.842	0.806	**0.892**
3.2.1.3	**0.898**	0.746	0.807	0.828
3.2.1.4	0.964	**0.969**	0.850	0.935
3.2.1.5	**0.932**	0.854	0.879	0.911
3.2.1.6	**0.912**	0.776	0.822	0.874
3.2.1.7	0.909	0.812	0.922	**0.962**
3.3.1.2	**0.937**	0.847	0.836	0.897
3.3.1.5	**0.917**	0.709	0.828	0.857
3.32.1.1	**0.946**	0.866	0.826	0.863
3.32.1.11	0.880	0.888	**0.947**	0.937
3.32.1.13	0.836	0.646	**0.901**	0.880
3.32.1.8	**0.901**	0.776	0.781	0.858
3.42.1.1	0.886	**0.923**	0.795	0.794
3.42.1.5	**0.811**	0.580	0.665	0.677
3.42.1.8	0.760	**0.930**	0.710	0.780
7.3.10.1	0.986	**0.997**	0.978	0.991
7.3.5.2	**0.996**	0.992	0.919	0.977
7.3.6.1	0.998	**0.999**	0.945	0.972
7.3.6.2	0.994	**0.997**	0.969	0.973
7.3.6.4	0.992	**1.000**	0.993	0.993
7.39.1.2	**0.928**	0.877	0.898	0.854
7.39.1.3	**0.990**	0.985	0.922	0.820
7.41.5.1	0.791	**0.916**	0.505	0.756
7.41.5.2	0.943	**0.999**	0.605	0.976
average	**0.893**	0.817	0.846	0.876
std. dev.	0.133	0.171	0.133	**0.105**
min	0.146	0.345	0.403	**0.493**

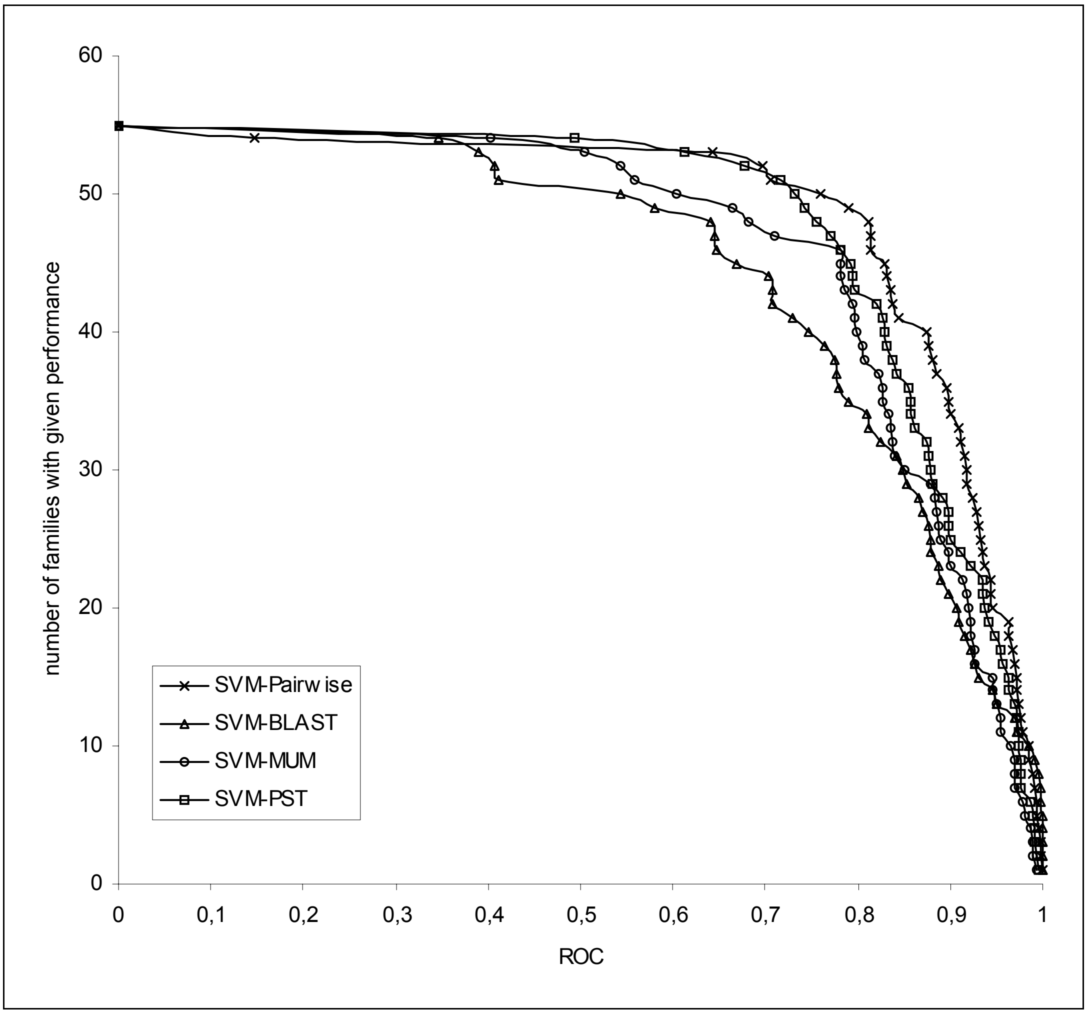


Figure 11. Relative performances of SVM methods based on the pairwise similarity scores that are depicted by the plots of the number of families for which a given method exceeds a threshold ROC score.

The results were also compared in family-by-family using pairwise comparison plots. As it is seen from the plots, the family classification performance of SVM-MUM is comparable to that of SVM-Pairwise for some of the SCOP families (Figure 12), while it is apparently better than SVM-Fisher (Figure 13). The plots of SVM-MUM vs. SVM-BLAST (Figure 14) justifies our above argument that MUM-based comparison provides a more accurate solution to eliminating twilight zone of sequence comparisons in comparison with BLAST.

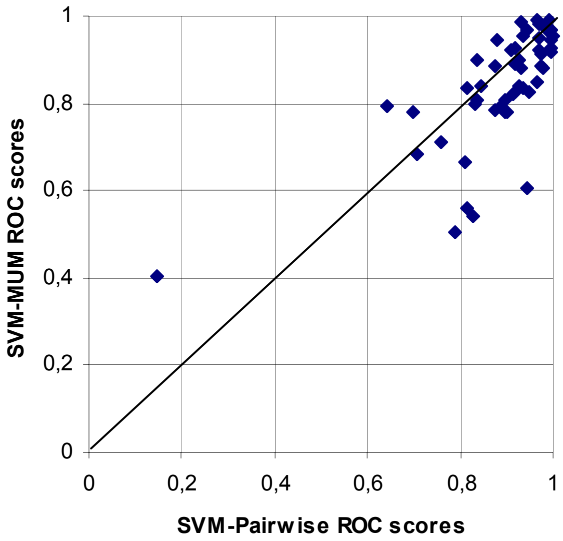


Figure 12. SVM-MUM vs. SVM-Pairwise

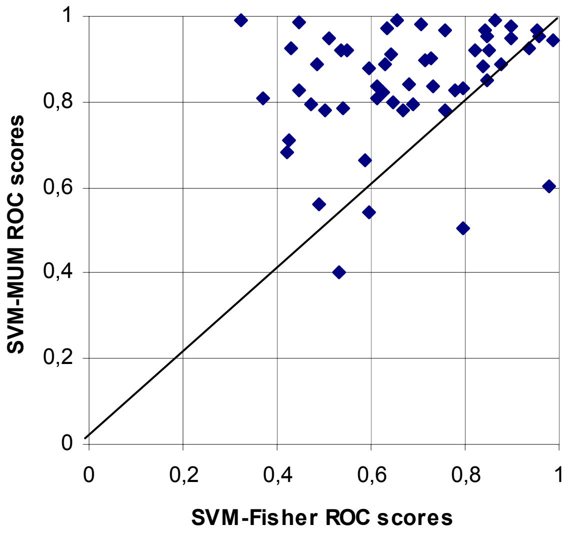


Figure 13. SVM-MUM vs. SVM-Fisher

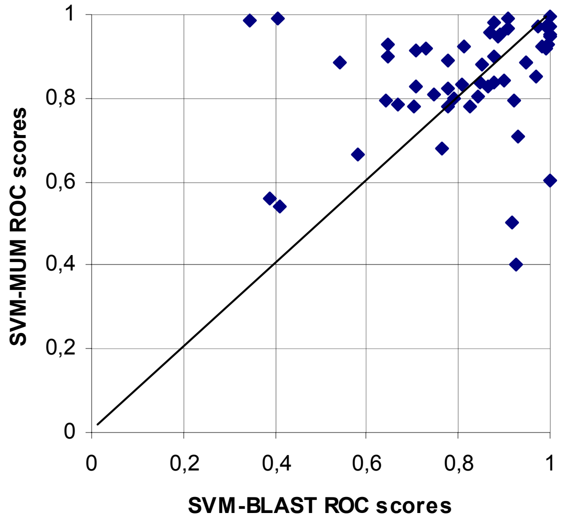


Figure 14. SVM-MUM vs. SVM-BLAST

Figure 15, Figure 16 and Figure 17 illustrate the family-by-family comparison of SVM-PST with SVM-Pairwise, SVM-Fisher and SVM-BLAST. As shown, SVM-PST performs better than SVM-Fisher for all families and better than SVM-BLAST for most of the families. An interesting point is observed in the comparison of SVM-PST and SVM-Pairwise for the family of eukaryotic proteases. The new method provides a remarkable improvement for this family.

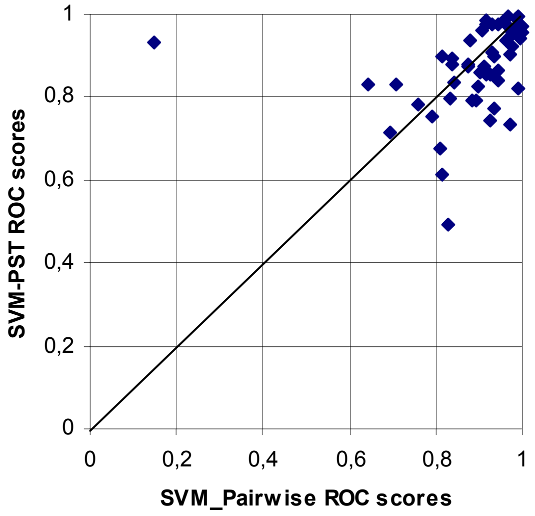


Figure 15. SVM-PST vs. SVM-Pairwise

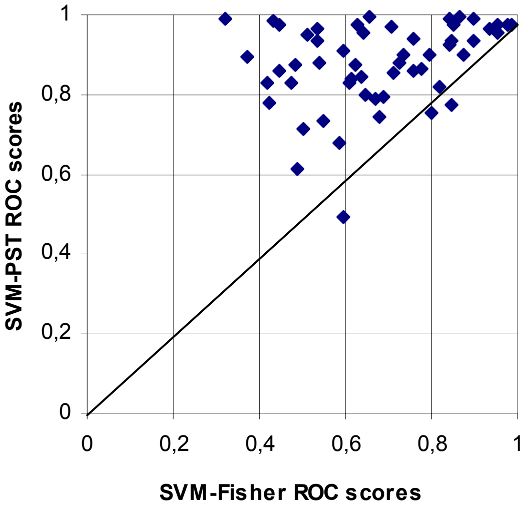


Figure 16. **SVM-PST vs. SVM-Fisher**

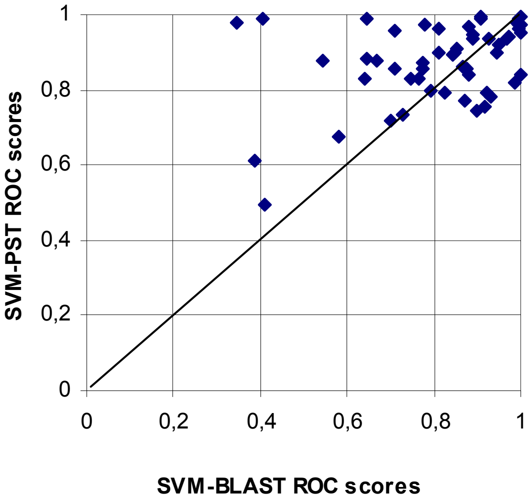


Figure 17. **SVM-PST vs. SVM-BLAST**

To compare the classification performance of SVM-PST with the original family-based PST approach, we constructed PSTs from multiple sequences in the training set of each family and produced prediction probabilities for each protein in the test set. The proteins were sorted by their resulting probabilities and the ROC scores were calculated for each family. The average ROC score obtained with this method was 0.508, which is far less than the score achieved by SVM-PST (*p*=2.41E-12). This emphasizes the superior performance of pairwise PSTs when combined with SVMs in comparison with the family-based PST approach.

The SVM performances for varying compositional encoding parameters are given in Table 8. As shown in the table, the system's performance increases until the inclusion of *n*-peptide composition with *n*=6 and degrades after *n* becomes larger than 8. With a threshold of 5000 in vector dimensions, the number of letters in the simplified alphabet for *n*>7 must be reduced to 2. Since an alphabet size of 2 does not carry any information about the protein evolution, this is possibly the reason for decrease in the accuracy after *n*>7. A comparison between the cases for different dimension thresholds applied is also presented in the table. According to the results, when the threshold is increased to 10000 no improvement is observed. On the other hand, the accuracy reduces when the threshold is lowered into 1000. Therefore, the threshold value of 5000 seems to be a good selection for satisfying both the accuracy and efficiency requirement of the system. The table also demonstrates the surprising success of amino acid composition (*n*=1) alone in homology detection task. The accuracy achieved with the use of amino acid composition is better than many of the more complicated methods included in our comparative study.

SVM-*n*-peptide performs better than SVM-Pairwise for almost half of the SCOP families used in the experiments. When the results are investigated in superfamily level, it is observed that SVM-*n*-peptide is more successful for all families in homeodomain-like proteins (1.4.1.x), nucleic acid-binding proteins (2.38.4.x), viral coat and capsid proteins (2.9.1.x), glycosyltransferases (3.1.8.x) and P-loop containing nucleotide triphosphate hydrolases (3.32.1.x). This result would be useful when selecting the appropriate method in any application that requires an automated or semi-automated search in SCOP database.

Table 8. Comparison of composition-based feature representations for dicriminative remote homology detection

				n=1-3	*n*=1-4	*n*=1-6	*n*=1-6	*n*=1-6	*n*=1-7	*n*=1-10
family	***n*=1**	***n*=2**	***n*=3**	***t*=5000**	***t*=5000**	***t*=1000**	***t*=5000**	***t*=10000**	***t*=5000**	***t*=5000**
1.27.1.1	0,829	0,914	0,731	0,931	0,952	0,891	0,948	0,938	0,947	0,930
1.27.1.2	0,931	0,973	0,852	0,969	0,967	0,937	0,962	0,958	0,957	0,928
1.36.1.2	0,755	0,837	0,543	0,868	0,842	0,910	0,916	0,841	0,908	0,874
1.36.1.5	0,660	0,934	0,467	0,873	0,873	0,917	0,961	0,920	0,961	0,951
1.4.1.1	0,990	0,970	0,818	0,993	0,987	0,924	0,977	0,986	0,984	0,976
1.4.1.2	0,916	0,907	0,913	0,949	0,958	0,919	0,976	0,977	0,982	0,962
1.4.1.3	0,969	0,838	0,869	0,974	0,967	0,984	0,973	0,987	0,974	0,942
1.41.1.2	0,781	0,989	0,959	0,976	0,991	0,998	0,994	0,996	0,994	0,992
1.41.1.5	0,924	0,965	0,963	0,988	0,993	0,985	0,990	0,991	0,993	0,989
1.45.1.2	0,869	0,897	0,445	0,951	0,936	0,708	0,810	0,949	0,926	0,876
2.1.1.1	0,700	0,833	0,888	0,864	0,905	0,900	0,892	0,909	0,903	0,888
2.1.1.2	0,904	0,943	0,973	0,968	0,976	0,987	0,985	0,989	0,985	0,978
2.1.1.3	0,903	0,964	0,946	0,984	0,985	0,966	0,976	0,981	0,981	0,971
2.1.1.4	0,760	0,835	0,867	0,869	0,895	0,866	0,894	0,895	0,909	0,895
2.1.1.5	0,685	0,640	0,832	0,711	0,768	0,784	0,807	0,809	0,831	0,808
2.28.1.1	0,496	0,670	0,335	0,685	0,723	0,415	0,637	0,570	0,644	0,662
2.28.1.3	0,832	0,563	0,743	0,756	0,767	0,802	0,865	0,822	0,860	0,855
2.38.4.1	0,810	0,538	0,496	0,581	0,672	0,792	0,766	0,738	0,756	0,866
2.38.4.3	0,823	0,769	0,532	0,820	0,773	0,755	0,779	0,736	0,739	0,750
2.38.4.5	0,844	0,831	0,666	0,842	0,861	0,816	0,916	0,869	0,886	0,821
2.44.1.2	0,296	0,394	0,296	0,231	0,215	0,300	0,259	0,203	0,187	0,167
2.5.1.1	0,869	0,901	0,697	0,921	0,933	0,854	0,896	0,918	0,899	0,824
2.5.1.3	0,745	0,755	0,729	0,797	0,805	0,759	0,783	0,784	0,772	0,767
2.52.1.2	0,564	0,718	0,852	0,822	0,786	0,714	0,783	0,762	0,792	0,738
2.56.1.2	0,849	0,743	0,682	0,888	0,893	0,911	0,855	0,929	0,851	0,800
2.9.1.2	0,919	0,950	0,771	0,953	0,948	0,938	0,951	0,940	0,951	0,956
2.9.1.3	0,993	0,998	0,970	0,999	0,997	0,995	0,996	0,998	0,998	0,996
2.9.1.4	0,983	0,985	0,918	0,992	0,992	0,983	0,984	0,989	0,983	0,975
3.1.8.1	0,969	0,973	0,942	0,990	0,991	0,992	0,987	0,993	0,987	0,973
3.1.8.3	0,967	0,956	0,964	0,978	0,980	0,967	0,973	0,985	0,976	0,956
3.2.1.2	0,718	0,790	0,774	0,832	0,847	0,875	0,887	0,853	0,886	0,868
3.2.1.3	0,755	0,744	0,720	0,773	0,783	0,841	0,859	0,817	0,875	0,872
3.2.1.4	0,712	0,929	0,902	0,941	0,940	0,943	0,939	0,939	0,938	0,940
3.2.1.5	0,885	0,879	0,860	0,896	0,920	0,911	0,914	0,909	0,915	0,909
3.2.1.6	0,779	0,763	0,833	0,866	0,891	0,886	0,903	0,901	0,892	0,888
3.2.1.7	0,852	0,942	0,979	0,966	0,956	0,957	0,955	0,949	0,940	0,913
3.3.1.2	0,774	0,818	0,788	0,895	0,885	0,804	0,916	0,913	0,934	0,929
3.3.1.5	0,704	0,900	0,811	0,897	0,910	0,900	0,943	0,933	0,948	0,931
3.32.1.1	0,888	0,910	0,909	0,932	0,954	0,933	0,952	0,945	0,946	0,940
3.32.1.11	0,846	0,944	0,948	0,975	0,989	0,982	0,973	0,983	0,968	0,947
3.32.1.13	0,659	0,882	0,829	0,886	0,905	0,907	0,938	0,950	0,949	0,949
3.32.1.8	0,764	0,810	0,874	0,877	0,915	0,914	0,912	0,902	0,906	0,899
3.42.1.1	0,706	0,686	0,576	0,755	0,839	0,780	0,840	0,817	0,846	0,830
3.42.1.5	0,599	0,713	0,480	0,673	0,712	0,594	0,624	0,631	0,648	0,636
3.42.1.8	0,758	0,688	0,650	0,690	0,734	0,675	0,674	0,713	0,696	0,682
7.3.10.1	0,987	0,984	0,970	0,991	0,991	0,981	0,985	0,991	0,987	0,973
7.3.5.2	0,926	0,987	0,990	0,978	0,992	0,981	0,987	0,994	0,992	0,992
7.3.6.1	0,953	0,966	0,988	0,990	0,994	0,997	0,978	0,995	0,973	0,971
7.3.6.2	0,808	0,951	0,952	0,962	0,967	0,979	0,965	0,975	0,966	0,967
7.3.6.4	0,996	0,979	0,985	0,996	0,996	0,994	0,995	0,998	0,998	0,993
7.39.1.2	0,979	0,917	0,794	0,938	0,904	0,910	0,863	0,908	0,776	0,778
7.39.1.3	0,820	0,835	0,931	0,829	0,813	0,792	0,870	0,846	0,836	0,833
7.41.5.1	0,832	0,847	0,577	0,841	0,826	0,800	0,841	0,825	0,830	0,822
7.41.5.2	0,744	0,982	0,683	0,981	0,953	0,931	0,860	0,957	0,909	0,914
average	0,814	0,851	0,786	0,879	0,888	0,869	0,890	0,889	0,890	0,879
std.dev	0,139	0,132	0,180	0,134	0,128	0,138	0,125	0,136	0,132	0,133
min	0,296	0,394	0,296	0,231	0,215	0,300	0,259	0,203	0,187	0,167

Figure 18 compares the performance of new method for *n*=1-6 and *t*=5000 with the existing methods using the plots of the number of families for which a given method exceeds a threshold ROC score.

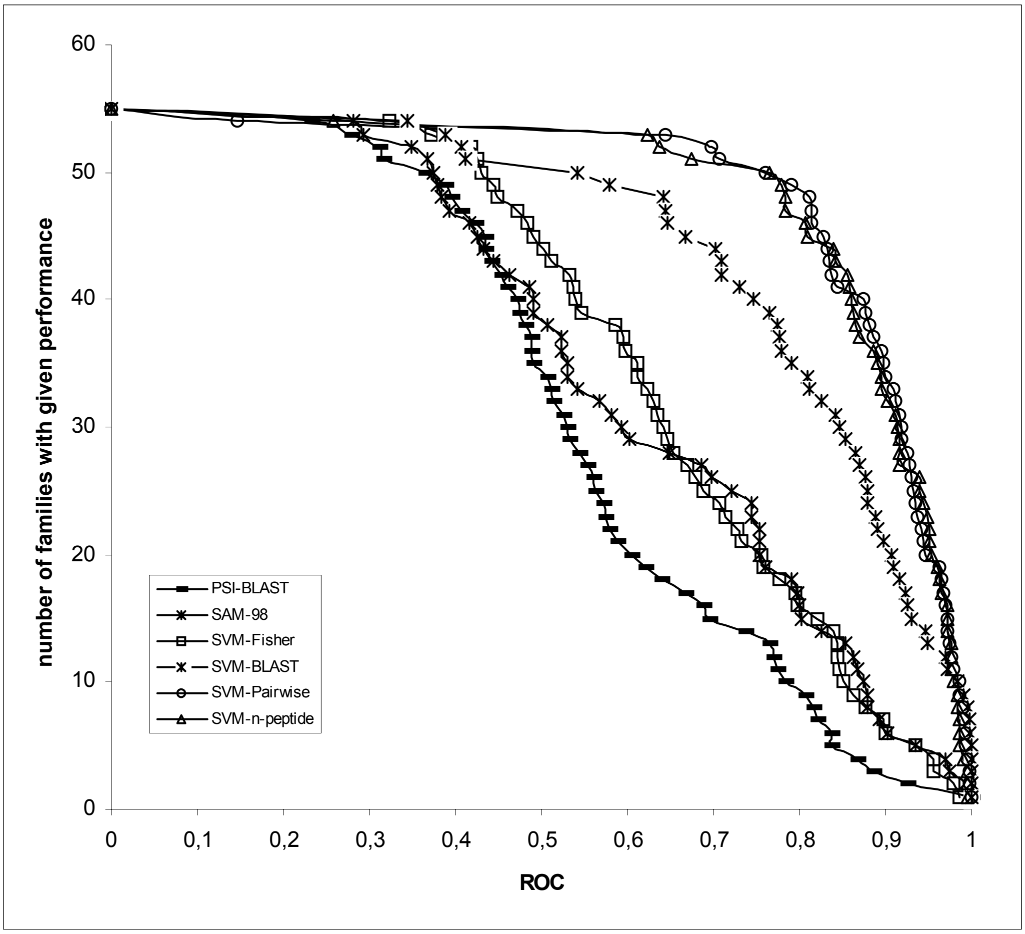


Figure 18. Relative performances of different classification methods that are depicted by the plots of the number of families for which a given method exceeds a threshold ROC score.

Family-by-family comparisons between SVM-n-peptide and SVM-Pairwise, SVM-Fisher and SVM-BLAST are also provided in Figure 19, Figure 20 and Figure 21. Both average ROC scores and the comparison plots demonstrate that the new method significantly outperforms all given methods except SVM-Pairwise, while being competitive and complementary for many of the SCOP families included in the experimental setup.

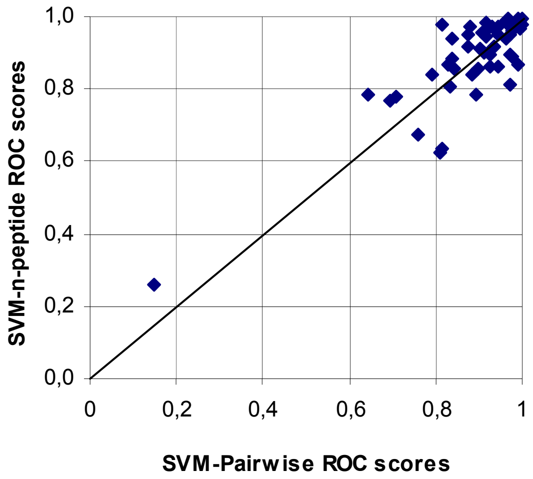


Figure 19. SVM-*n*-peptide vs. SVM-pairwise

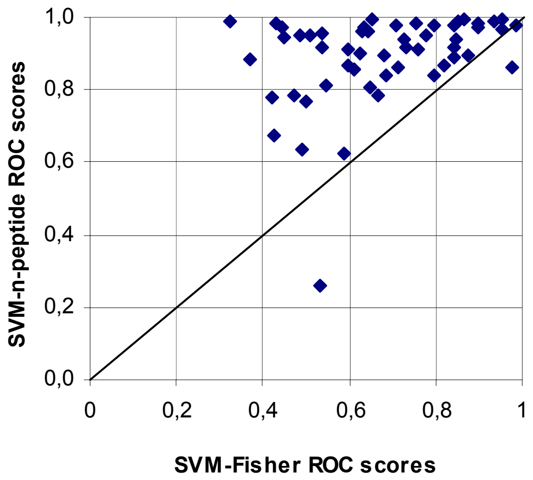


Figure 20. SVM-*n*-peptide vs. SVM-Fisher

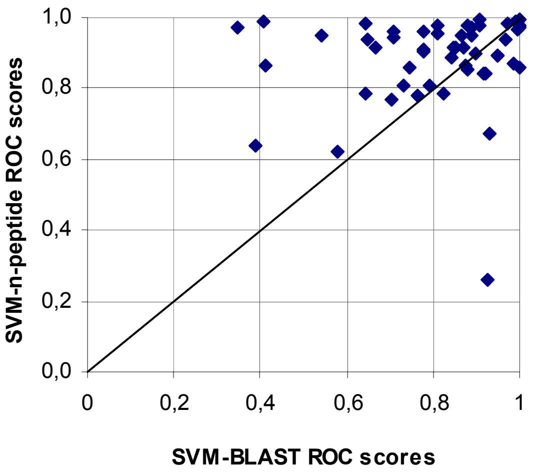


Figure 21. **SVM-*n*-peptide vs. SVM-BLAST**

In order to see the effect of different amino acid groupings, we also applied the schemes provided by Li, Fan, Wang and Wang (2003) and Liu, Liu, Qi and Zheng (2002) for reducing alphabets. The results are shown in Table 9. For the homology detection tests with *n*=1-6 and *t*=5000, former scheme provided an average ROC score of 0.893 with 0.132 standard deviation and the latter one provided 0.889 average ROC score with 0.134 standard deviation. Although we do not observe a statistically significant difference between them (*p*>0.05 for all paired T-tests), the ROC score deviations with the alphabets of Murphy et al. is lower than those with other alphabets.

When the new methods are compared to each other by pairwise comparison plots, it is observed that SVM-*n*-peptide performs slightly better than SVM-PST (Figure 19), while both achieve a greater accuracy than SVM-MUM for most of the SCOP families (Figure 20, Figure 21). The reasons for the superiority of SVM-*n*-peptide to SVM-MUM are likely the possibility of evaluating certain mismatches in searched substrings and inclusion of higher number of motifs in the sequence feature representation.

Table 9. Comparison of composition-based feature representations (*n*=1-6 *t*=5000) with different amino acid grouping schemes

family	**Murphy et al.**	**Li at al.**	**Liu et al.**
1.27.1.1	0,948	0,945	0,948
1.27.1.2	0,962	0,967	0,985
1.36.1.2	0,916	0,862	0,842
1.36.1.5	0,961	0,937	0,920
1.4.1.1	0,977	0,986	0,987
1.4.1.2	0,976	0,958	0,931
1.4.1.3	0,973	0,989	0,988
1.41.1.2	0,994	0,998	0,995
1.41.1.5	0,990	0,992	0,997
1.45.1.2	0,810	0,972	0,938
2.1.1.1	0,892	0,903	0,927
2.1.1.2	0,985	0,991	0,980
2.1.1.3	0,976	0,989	0,987
2.1.1.4	0,894	0,913	0,892
2.1.1.5	0,807	0,809	0,743
2.28.1.1	0,637	0,647	0,551
2.28.1.3	0,865	0,783	0,811
2.38.4.1	0,766	0,703	0,652
2.38.4.3	0,779	0,764	0,751
2.38.4.5	0,916	0,826	0,830
2.44.1.2	0,259	0,202	0,283
2.5.1.1	0,896	0,912	0,926
2.5.1.3	0,783	0,807	0,810
2.52.1.2	0,783	0,809	0,862
2.56.1.2	0,855	0,894	0,949
2.9.1.2	0,951	0,939	0,927
2.9.1.3	0,996	0,996	0,999
2.9.1.4	0,984	0,989	0,985
3.1.8.1	0,987	0,994	0,985
3.1.8.3	0,973	0,983	0,982
3.2.1.2	0,887	0,847	0,830
3.2.1.3	0,859	0,843	0,823
3.2.1.4	0,939	0,950	0,954
3.2.1.5	0,914	0,908	0,882
3.2.1.6	0,903	0,884	0,882
3.2.1.7	0,955	0,964	0,950
3.3.1.2	0,916	0,911	0,938
3.3.1.5	0,943	0,931	0,940
3.32.1.1	0,952	0,946	0,939
3.32.1.11	0,973	0,991	0,981
3.32.1.13	0,938	0,964	0,942
3.32.1.8	0,912	0,891	0,926
3.42.1.1	0,840	0,772	0,826
3.42.1.5	0,624	0,643	0,583
3.42.1.8	0,674	0,780	0,789
7.3.10.1	0,985	0,991	0,992
7.3.5.2	0,987	0,993	0,996
7.3.6.1	0,978	0,993	0,996
7.3.6.2	0,965	0,977	0,973
7.3.6.4	0,995	1,000	1,000
7.39.1.2	0,863	0,896	0,965
7.39.1.3	0,870	0,870	0,778
7.41.5.1	0,841	0,862	0,818
7.41.5.2	0,860	0,975	0,991
average	0,890	0,893	0,890
std.dev	**0,125**	**0,132**	**0,134**
min	**0,259**	**0,202**	**0,283**

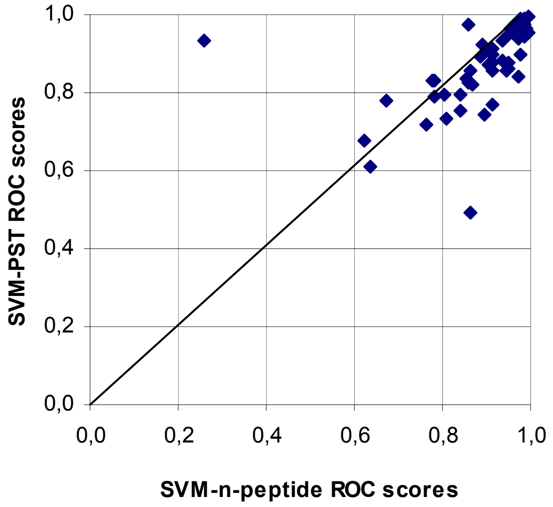


Figure 22. SVM-PST vs. SVM-*n*-peptide

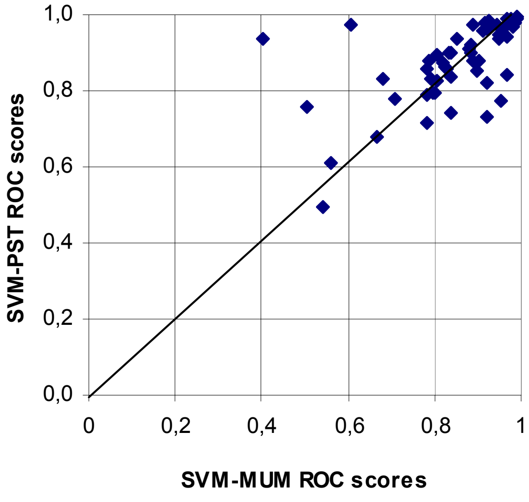


Figure 23. SVM-PST vs. SVM-MUM

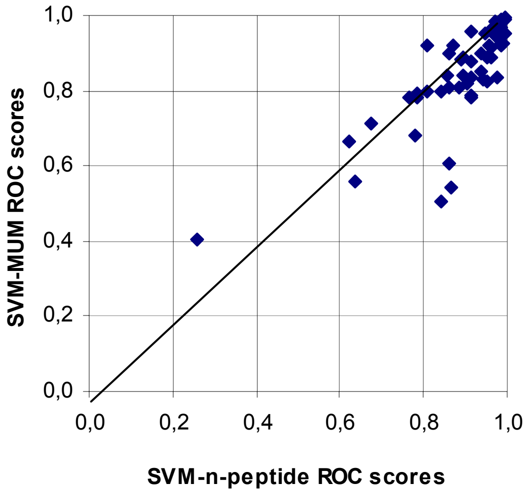


Figure 24. SVM-MUM vs. SVM-*n*-peptide

To explore the statistical significance of differences between the results, paired-samples T-tests were carried out between SVM-based methods with a *p*-value threshold of 0.05. According to Table 10, the difference between SVM-*n*-peptide and SVM-BLAST and that between SVM-PST and SVM-BLAST are statistically significant. The difference between SVM-*n*-peptide and SVM-Fisher and the one between SVM-PST and SVM-Fisher are more apparent. On the other hand, no significant difference is observed between SVM-*n*-peptide, SVM-PST and SVM-Pairwise.

Table 10. Paired-samples T-test results for comparison of SVM-based remote homology detection methods

p-value	SVM-n-pep.	SVM-PST	SVM-MUM	SVM-BLAST	SVM-Fisher
SVM-pairwise	3,70E-01	3,83E-01	9,88E-04	5,41E-03	6,62E-13
SVM-n-peptide		3,94E-01	3,96E-03	5,00E-03	3,90E-12
SVM-PST			5,08E-02	8,55E-03	2,70E-11
SVM-MUM				2,64E-01	2,92E-08
SVM-BLAST					7,70E-10
SVM-Fisher					-

Computational efficiency

Computational efficiency is another important aspect in the evaluation of methods. The crucial step in the SVM system we used is the vectorization of proteins. The time complexities are given in Table 11.

Table 11. Time complexities of discriminative remote homology detection methods in vectorization step

Method	Time complexity
SVM-Fisher	$O(mp)$
SVM-BLAST	$O(km)$
SVM-Pairwise	$O(km^2)$
SVM-MUM	$O(km)$
SVM-PST	$O(kLm)$
SVM-*n*-peptide	$O(m)$

The vectorization step has a complexity of $O(mp)$ in SVM-Fisher, where m is the length of the longest training set sequence and p is the number of parameters used in the profile HMM. SVM-Pairwise and SVM-BLAST calculate all pairwise similarity scores between the target sequence and the sequences in the training set. Each similarity calculation is $O(m^2)$ in SVM-Pairwise and $O(m)$ in SVM-BLAST. Thus, total vectorization time is $O(km^2)$ for SVM-Pairwise and $O(km)$ for SVM-BLAST, where k is the number of proteins in the training set. On the other hand, all MUMs can be identified in $O(m)$ time, which constitutes a $O(km)$ time complexity in the vectorization phase of SVM-MUM. PST comparison is completed in two stages; while the PST construction for the first sequence is completed in $O(m)$ time, the prediction of the second sequence also takes $O(m)$ time (Apostolico and Bejerano, 2000). With our modification that maximizes the probability scores for each residue, the complexity of the prediction step increase to $O(Lm)$ for the worst case. Then, the complexity of SVM-PST in vectorization phase becomes $O(kLm)$. SVM-*n*-peptide constructs *n*-peptide compositions for feature vectorization. This scheme has a time complexity of $O(m)$ as described in Section 2.1. The complexity analysis reveals that SVM-*n*-peptide is the most efficient system among all compared methods. This result is mainly due to that compositional representation of a sequence is independent from the size of the training set. According to the analysis, the complexities of SVM-MUM, SVM-PST, SVM-BLAST and SVM-Fisher are in the same order, while all of them are more efficient than SVM-Pairwise.

An empirical comparison in terms of computation time may be invalid since much of the work in our implementation contains file processing owing to large amount of data that can not be handled by memory. However, to make an intuition, we can report that all training time is at most one hour for a family with SVM-*n*-peptide, whereas it takes at least 20 days with SVM-Pairwise in a workstation having 1GHz CPU and 1GB memory. SVM-MUM completes all training stage in 4 days on average, while SVM-PST performs similarly and can complete the training in 5 days.

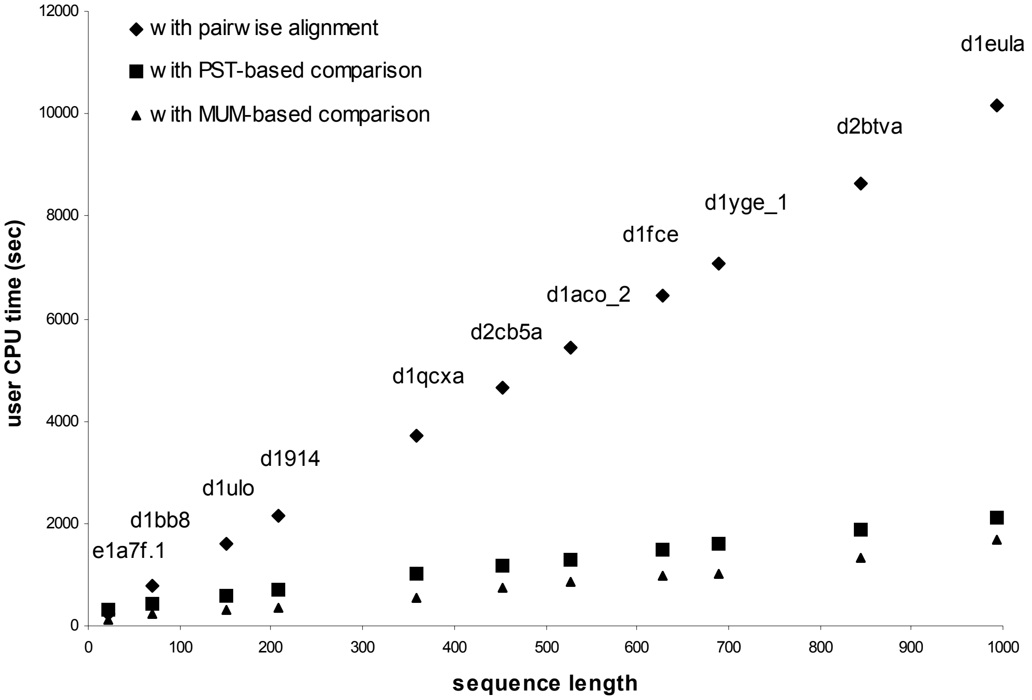


Figure 25. Computation times for protein vectorization with pairwise alignment scores, PST-based similarity scores and MUM-based scores.

To make an empirical comparison in terms of testing time, we randomly selected some proteins with varying lengths between 20 and 1000 and measured the user CPU times (in seconds) spent for the vectorization of these proteins. We did not use the training set of a specific family, but rather used all proteins in the data set to vectorize a test protein. In other words, the vectorization mentioned for this performance test refers to computing the similarity scores between the test protein and all other proteins in the data set. The computation times are given in Figure 25 with the associated protein labels and their lengths (SVM-n-peptide measurements are not shown because it requires much less time). As illustrated in the figure, the vectorization time increases more drastically with the sequence length when the dynamic-programming-based alignment scores are used. PST-based and MUM-based comparisons become apparently more advantageous when the sequence length gets higher than 100. Since the SVM-related computations require much less time (only a few seconds for testing one protein) than the vectorization process, we can conclude that the SVM-*n*-peptide, SVM-PST and SVM-MUM methods substantially reduce the computation time needed for both training and testing phases while mostly preserving the accuracy achieved by SVM-Pairwise.

CHAPTER 4

Subcellular Localization Prediction

When the translation process completes, a protein may reside in several locations in the cell. The subcellular localization of a protein is one of the key characteristics in elucidating its function. The knowledge of protein localization may provide valuable information in the target identification process for drug discovery. Owing to the fact that the experimental analysis of subcellular localization is a costly and time consuming process, developing automated tools for classifying proteins into their subcellular localization sites has become increasingly important in recent years.

This study aims to show the applicability of new sequence representations in subcellular localization prediction problem. In addition to the generic representations introduced in earlier chapters, some problem-specific modifications are also proposed. Finally, a hybrid system that combines several sequence representation schemes is presented. The new system, called as PredLOC, is compiled for the prediction of subcellular location in eukaryotic proteins and shown to outperform all existing methods in the experiments carried on two distinct benchmarking data sets, which are based on four experimentally characterized subcellular localization.

4.1 Previous Studies

Various methods have been introduced for predicting subcellular localization of proteins. Some of the methods are based on the existence of targeting signals appearing in N-terminal sequences (PSORT-Horton and Nakai, 1999, TargetP-Emanuelsson et al., 2000). By the identification of localization specific targeting signals, a statistical analysis is performed to predict subcellular localization. Two main problems exist with this approach. First, the reliability of predictions is highly correlated with the type of the proteins to be predicted. The methods require a priori information about which kind of signals are to be searched for the given type of organisms and their specific localization sites. The second problem is the fact that the non-existence of certain signals in the target protein makes it difficult to classify the protein into given localizations, which reduces the coverage of the prediction system. The PSORT method was later improved by adding some other analytical modules each of which analyzes a distinct feature known to influence or be characteristic of subcellular localization (PSORTB-Gardy et al., 2003; PSORTII-Gardy et al., 2005). Other methods are based on the use of machine learning classifiers with compositional features of the protein sequences. Nakashima and Nishikwa (1994) showed that the amino acid composition of a protein is an important feature in the determination of its being intracellular or extracellular. Reinhard and Hubbard used the amino acid composition to train neural networks for classifying proteins into known subcellular localizations (NNPSL-Reinhard and Hubbard, 1998). Later, same feature representation scheme was used with SVM (SubLoc-Hua and Sun, 2001) and fuzzy k-NN (Huang and Li, 2004). Dipeptide composition, biochemical properties and PSI-BLAST homology search results were additionally used to train the SVMs in ESLPred (Bhasin and Raghava, 2004) and PSLPred (Bhasin, Garg and Raghava, 2005) systems. Similar idea was used in the development of LOCSVMPSI (Xie, Wang, Fan and Feng, 2005), which integrates PSSM (Position Specific Scoring Matrix- Gribskov et al., 1990) output of homology searches instead of binary search results into SVM. Park and Kanehisa (2003) used a voting scheme between different SVM predictions based on amino acid composition, dipeptide composition and gapped pair composition for final prediction. Nair and Rost (2003) used neural networks in the LOCnet system with predicted 1D structure information and evolutionary profiles. A new kernel based on implicit motif distribution were used with SVMs in P2SL (Atalay and Atalay, 2005). SVM-based methods are currently among the best methods in terms of prediction accuracy. They differ only on the protein encoding schemes they have used.

4.2 Systems and Methods

Two novel sequence representation schemes that are introduced in Chapter 2 are used in the SVM-based subcellular localization prediction framework. The first representation scheme uses *n*-peptide compositions of proteins with reduced amino acid alphabets for larger values of *n*. The second one is based on the pairwise similarity scores between the target protein and the other proteins in the training set, in which the pairwise similarity scores are calculated using new probabilistic suffix tree model described in Section 2.2.3. We also developed a hybrid system; we called as PredLOC, which combines the results of the distinct SVMs based on the two encoding schemes presented.

4.2.1 Multi-class Prediction with SVMs

As described in Chapter 3, SVM is a binary classifier; that is, it can separate between two classes. The SVM output is a discriminant score corresponding to the test sample to be classified. In a binary classification task, a positive value of this score indicates that the test sample belongs to that class. For an *N*-class problem, a separate support vector machine SVM_i is constructed for each class *i*, where *i* is between 1 and *N*. In the prediction phase, a discriminant score of d_i, which is obtained from the SVM_i, is assigned to the test sample. There are several methods for evaluating *N* discriminant scores and selecting most appropriate class for the test samples. The most common solution is selecting the class for which the highest score is attained. Since the problem of predicting subcellular localization is a multi-class problem, we adopt this solution for the assignment of proteins into most appropriate localization site. That is, a separate support vector machine, *SVM_i*, is constructed for each localization site *i* in the data set, and the target protein is categorized into the localization site *x* for which the test protein attains the maximum discriminant value, d_x, from its *SVM*$_x$. An overview of the system is illustrated in Figure 26.

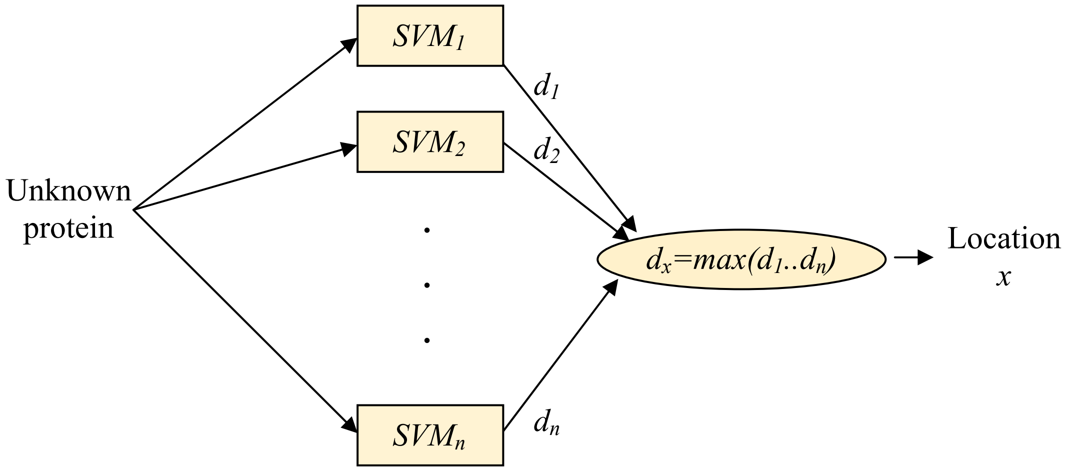


Figure 26. Multi-class SVM prediction system for subcellular localization

4.2.2 Feature Representations

In the first scheme, we represent proteins by their *n*-peptide compositions. Owing to the lessons learned from the results of remote homology detection experiments, the parameters for *n*-peptide composition construction for this problem are selected as *n*=1-6 and *t*=5000. The reduced amino acid alphabet set given by Murphy et al. (2000) is used since it provided the most reliable classification in remote homology detection tests.

The sequence representation based on pairwise similarity scores is employed as the second method for feature vectorization. For this scheme, it was already stated that the alignment scores are most accurate way of construction pairwise similarity scores in the SVM-Pairwise framework for family classification task. By two reasons, we adopt the new definition for sequence similarity based on the pairwise PSTs for subcellular localization prediction problem. First, it is obvious that the alignment with dynamic programming is an inefficient way of measuring similarity; calculating all pairwise similarity scores would take several days in a conventional workstation. Second, the alignment does not allow changing the order of segments and may force some residues to match even if they are unrelated in localization point of view. Thus, a new method should have been provided which addresses the problem of measuring the level of significant matches regardless of their relative positions in the sequences. In this respect, the PST-based sequence similarity model is considered to be well-suited for this application.

Since the subcellular localization of a protein has been shown to be closely related to its N-terminal sequence for most of the localization sites, we also incorporate the similarity scores between the fixed-length N-terminal parts of the protein sequences to be compared. For convenience, we will use PST_{whole} for the encoding formed by the comparison of whole sequences and PST_K for the cases only first K letters of sequences are used for comparison.

4.2.3 Implementations and Experimental Setup

Same implementations with remote homology detection experiments were used for building *n*-peptide compositions. For PST-based comparisons, we ignored the two-way similarity approach proposed for distantly related sequences in order to reduce the memory requirements. A short memory length L of 7 was used in the PST construction and other parameters were adjusted so that all substrings shorter than L are represented by a node in the tree. SVM-Gist software was employed with the same parameters described in Section 3.1.

The data set we used contains 2427 eukaryotic proteins, which were extracted from SWISSPROT release 33.0 (Bairoch and Apweiler, 1999), with no pairwise sequence identity of more than 90% between them. The proteins are classified into four experimentally determined localization sites: nuclear, cytoplasmic, mitochondrial and extracellular. The numbers of proteins in each localization are 1097, 684, 321 and 325 respectively. This data set was constructed by Reinhardt and Hubbard for developing and testing their neural-network-based prediction system NNPSL and used later in the development of many methods such as SubLoc and ESLPred. We also used a new-unique proteins set, from SWISSPROT release 40 and 41, provided by Nair and Rost (2003) for testing only. This data set contains 512 samples having 146 cytoplasmic, 128 extracellular, 60 mitochondrial, and 178 nuclear proteins. This data set is compromised to be a more difficult set since it contains proteins with less identity which thus eliminating possible biases in the predictions.

The performance of the methods was evaluated through 5-fold cross validation test. In 5-fold cross-validation tests, 4/5 of the data set is chosen for training, and the remaining samples are used to evaluate the accuracy of the method. This evaluation is repeated five times by changing the partitions until all of the samples are run through a prediction process. The results were evaluated by accuracy measures and Matthew's correlation coefficients (Matthew, 1975) using the following equations:

$$Acc(i) = \frac{N_{ii}}{\sum_{j=1}^{j=c} N_{ij}} \qquad (4.1)$$

$$OverallAcairacy = \frac{\sum_{i=1}^{i=c} N_{ii}}{n} \qquad (4.2)$$

$$MCC(i) = \frac{p_i \cdot n_i - u_i \cdot o_i}{\sqrt{(p_i + u_i)(p_i + o_i)(n_i + u_i)(n_i + o_i)}} \qquad (4.3)$$

where, N_{ij} is the number of proteins in location i and predicted in location j, c is the number of classes (i.e. distinct locations), n is the number samples, p_i is the number of correctly predicted proteins in location i, n_i is the number of correctly predicted proteins not in location i, u_i is the number of under-predicted proteins and o_i is the number of over-predicted proteins. That is;

$$p_i = N_{ii} \tag{4.4}$$

$$n_i = \sum_{j=1, j\neq i}^{j=c} \sum_{k=1, k\neq i}^{k=c} N_{jk} \tag{4.5}$$

$$o_i = \sum_{j=1, j\neq i}^{j=c} N_{ji} \tag{4.6}$$

$$u_i = \sum_{j=1, j\neq i}^{j=c} N_{ij} \tag{4.7}$$

It is also important to know the prediction reliability when machine learning methods are used for subcellular localization. One of the common methods to evaluate the reliability is to use a reliability index (RI) which is calculated using the difference between the largest and second largest output of the system. Since an SVM system produce a single output, we can consider the difference between the discriminant scores produced by distinct SVMs devised for each localization site. Then, RI is calculated by the formula;

$$\text{RI=} \begin{cases} \text{INTEGER}(\mathit{diff}\text{*}5/2) \quad \text{, if } \mathit{diff}\text{<}5 \\ \\ 5 \text{ , if diff>=5} \end{cases} \tag{4.8}$$

where $diff = d_{max}$-$d_{secondmax}$.

The RI assignment provides useful information about the level of certainty in the prediction for a certain sequence. In other words, one can evaluate the prediction output of submitted sequence at which degree he can be confident about the result. After the experiments were completed, the prediction reliability of the proposed SVM system with new representation schemes were evaluated with plots of accuracy vs. RI and accuracy vs. percentage of predicted sequences with given RI.

4.3 Results and Discussion

In Table 12, the accuracies achieved by the SVM predictions based on *n*-peptide compositions are compared with the results of previous encoding schemes using amino acid composition, dipeptide composition, biochemical properties and their combination. As the table depicts, the new encoding scheme provided better accuracy than other representations for all localization sites.

Table 12. Comparison of SVM prediction accuracies for different encoding schemes for 2427 proteins set through 5-fold cross-validation tests.

Location	Amino acid composition (a)	Dipeptide composition (b)	Biochemical properties (c)	Combination of a, b and c	*n*-peptide compositions
Nuclear	86.1	92.7	85.6	93.2	94.3
Cytoplasmic	76.9	80.2	74.6	80.6	84.5
Mitochondrial	55.5	58.8	59.2	65.1	66.4
Extracellular	76.0	79.0	76.6	83.4	88.9
Overall	78.1	82.9	78.8	84.6	87.1

Results of (a),(b), and (c) were obtained from Bhasin and Raghava (2004). They used 33 different biochemical properties to obtain the representation in (c).

The prediction results with the pairwise similarity representations by PST_{20}, PST_{30}, PST_{40}, PST_{50}, PST_{whole} and some of their combinations are given in Table 13. When used with separate SVMs, PST_{whole} provided a higher overall accuracy compared with all PST_K encoding schemes, while the accuracies for individual locations might vary. The most notable result from the table is that nuclear proteins could be predicted more successfully by the PST_{whole} encoding in comparison with PST_{20}, PST_{30}, PST_{40} and PST_{50}. This result is consistent with the argument that nuclear proteins are destined by the specific motifs that may occur anywhere in the sequence (Weis, 1998; Cokol, Nair and Rost, 2000). It was also observed that the prediction accuracy increased when N-terminal and whole sequence similarity scores were combined into a single feature vector. Although PST_{50} and PST_{40} performed slightly better than PST_{30} in the individual tests, PST_{30} and PST_{40}

provided the same level of improvement when combined with PST_{whole}. The combination vector of PST_{50} and PST_{whole} achieved less accuracy. These observations are most likely due to the fact that PST_K and PST_{whole} carry almost the same information when K is increased, for especially short sequences.

Table 13. Comparison of PST-comparison-based SVM prediction accuracies for different N-terminal sequence lengths used for similarity calculation.

Location	PST_{20}	PST_{30}	PST_{40}	PST_{50}	PST_{whole}	PST_{20} + PST_{whole}	PST_{30} + PST_{whole}	PST_{40} + PST_{whole}	PST_{50} + PST_{whole}
Nuclear	76.7	80.9	81.9	82.7	92.0	89.6	90.5	90.6	90.6
Cytoplasmic	68.0	69.1	72.8	70.9	74.6	82.0	83.6	82.5	66.8
Mitochondrial	78.2	81.6	85.4	86.3	83.8	85.4	87.9	88.6	88.8
Extracellular	86.2	86.8	86.8	87.1	73.2	88.0	87.7	89.1	87.7
Overall	75.7	78.5	80.4	80.4	83.6	86.7	87.8	87.8	83.3

When we compare the new methods to each other, we observe that *n*-peptide composition and PST encoding (with PST_{30}+PST_{whole}) provided same overall accuracy whereas they showed disparate performance for individual locations. *n*-peptide composition achieved a higher accuracy for nuclear proteins possibly due to the same reason that is explained above for the difference between PST_K and PST_{whole}. On the other hand, the PST encoding is absolutely superior to compositional representations in predicting mitochondrial proteins. For cytoplasmic and extracellular proteins they perform nearly although the method based on *n*-peptide composition is slightly better.

For final prediction system, called PredLOC, we devised a decision rule that is based on the results of two independent SVM sets; SVM^1 and SVM^2, where SVM^1 uses *n*-peptide compositions and SVM^2 uses the combination vector of PST_{30} and PST_{whole}. The final decision is made over the average of discriminant scores obtained from SVM^1 and SVM^2. This system achieved a 91.3% overall accuracy for 2427 proteins-set through 5-fold cross-validation tests. According to the PredLOC test results, pi, ni, oi and ui values and accuracies for individual localizations are given in Table 14.

Table 14. PredLOC performance on 2427 proteins set

Location	p_i	n_i	o_i	u_i	Accuracy (%)
Nuclear	1041	1240	90	56	94.9
Cytoplasmic	595	1654	89	89	87.0
Mitochondrial	277	2085	21	44	86.3
Extracellular	301	2089	13	24	94.9

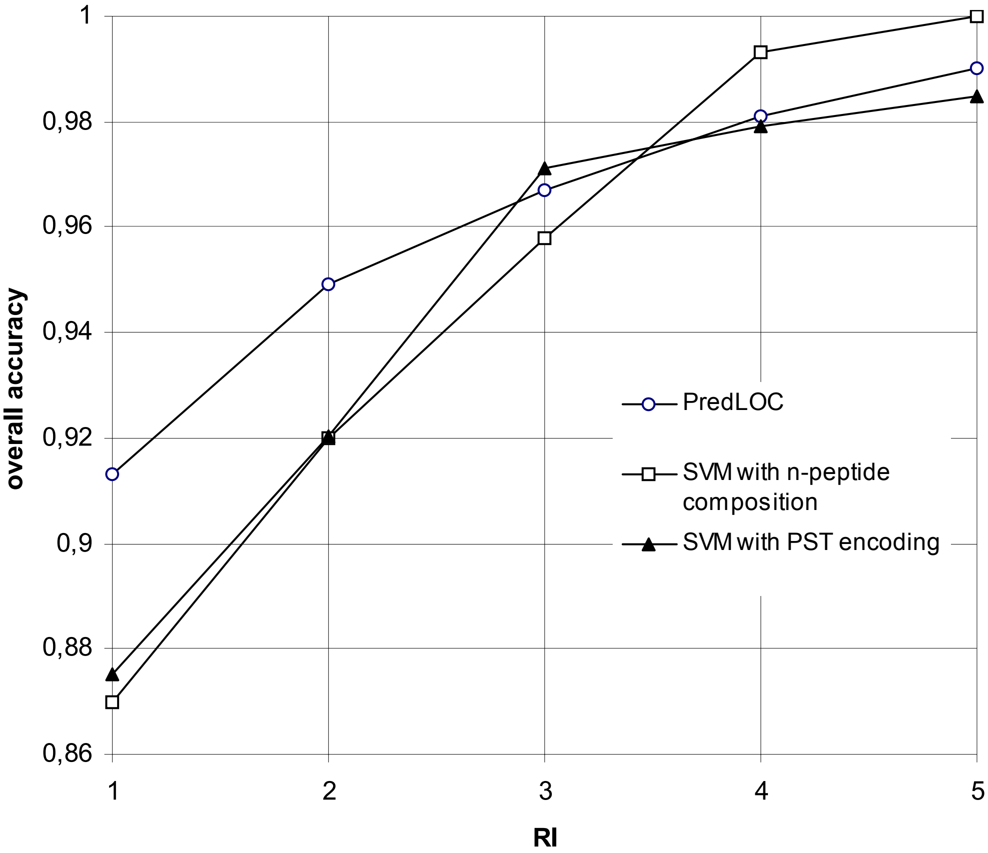


Figure 27. Expected localization prediction accuracy with a reliability index (RI) that is greater than a given value

To evaluate the reliability of the methods, two plots are presented. In Figure 27, overall accuracy measurements are plotted with respect to given RI values. Figure 28 gives the prediction accuracy of the methods versus the percentage of sequences that are predicted with a threshold reliability index. The prediction with *n*-peptide composition is more sensitive for higher RI, whereas its coverage falls for lower RI values. With PredLOC system, nearly 75% of the proteins have RI>=3, and expected accuracy for these proteins is 96.7%. As another result, approximately 40% of the sequences have RI>=5 and they can be predicted with 99% accuracy.

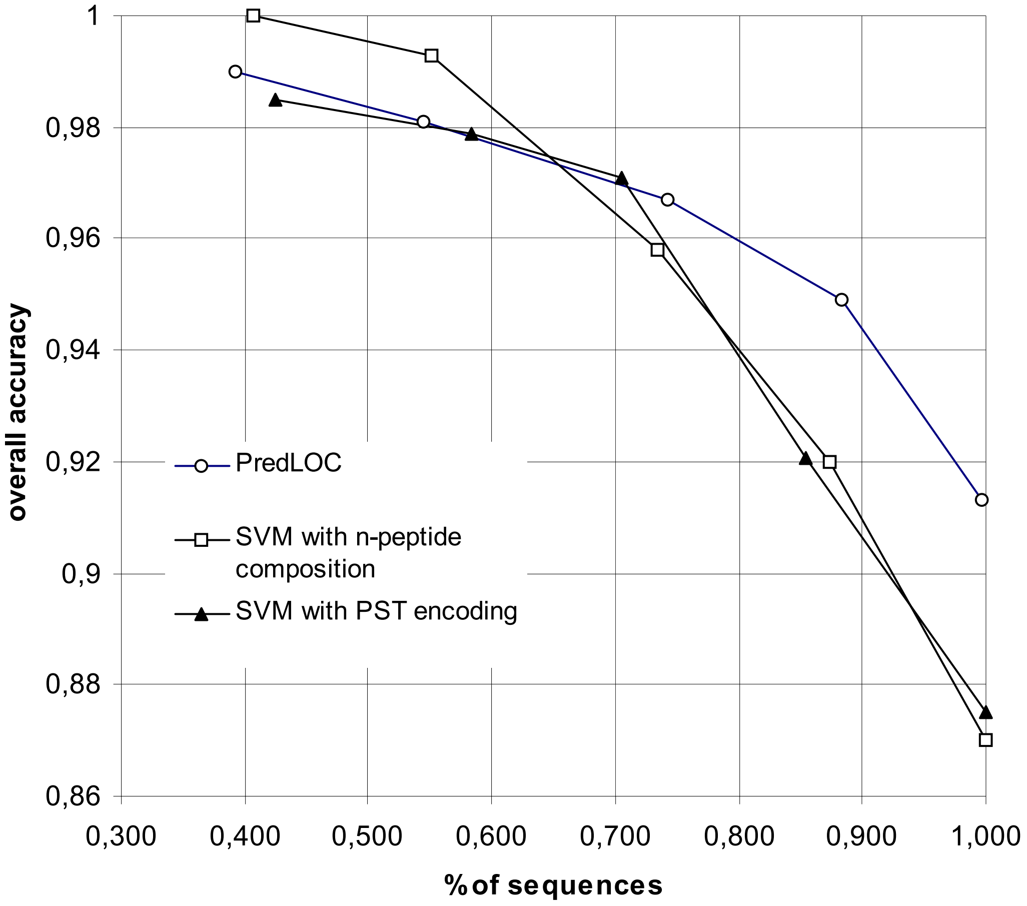


Figure 28. Expected localization prediction accuracy vs. the percentage of predicted sequences with a given RI.

Comparison with other methods

SubLoc reported an accuracy of 79.4% on the same dataset with a jackknife test. In the 5-fold cross-validation test results (Table 12), overall SVM prediction accuracy with amino acid composition, which is the method used by SubLoc, was 78.5%. In either case, PredLOC achieved nearly 12% greater accuracy than SubLoc. We presented a detailed comparison with ESLPred, which was tested on the same data set with 5-fold cross-validation procedure (Table 15). As seen from the table, PredLOC performed better than ESLPred for all localization sites except nucleus, with a 3.3% improvement in overall accuracy. A significant improvement was observed in prediction of mitochondrial proteins. The MCCs obtained from PredLOC were higher than those of ESLPred for all locations.

Table 15. Comparison of PredLOC with ESLPred on 2427 eukaryotic proteins through 5-fold cross-validation tests.

	ESLPred		PredLOC	
Location	Accuracy	MCC	Accuracy	MCC
Nuclear	**95.3**	0.87	94.9	**0.88**
Cytoplasmic	85.2	0.79	**87.0**	**0.83**
Mitochondrial	68.2	0.69	**86.3**	**0.88**
Extracellular	88.9	0.91	**94.9**	**0.94**
Overall	88.0	-	**91.3**	-

We also evaluated the performance of the new system on the new-unique SWISSPROT set by using all 2427 proteins as the training set. The prediction results were compared with many of the existing methods (Table 16). As shown, PredLOC performed better than all other methods in terms of overall accuracy. For individual locations, PredLOC performed best except extracellular location, for which LOCnet achieved a higher accuracy than PredLOC.

For the new-unique test set, three encoding schemes achieved different accuracies. With *n*-peptide compositions, the system achieved 69.2, 64.1, 70.0 and 79.8% accuracies for cytoplasmic, extracellular, mitochondrial and nuclear proteins respectively, and an overall accuracy of 71.7%. PST_{whole} encoding provided 57.5, 32.8, 81.7 and 82.6% for individual locations and 62.9% overall accuracy, where PST_{30} achieved a greater overall accuracy with 67.9% and provided 58.2, 73.4, 81.7

and 67.4% individual accuracies. Similar to the results in previous data set, PST_{whole} encoding is more successful in nuclear proteins, while it does not perform well in the detection of extracellular proteins due to the lack of homolog proteins in the training set.

Table 16. Comparison of prediction results on new-unique SWISSPROT test set.

	Accuracy				
Method	Overall	Nuclear	Cytoplasmic	Mitochondrial	Extracellular
NNPSL*	51.5	58	40	68	62
SubLoc*	57.4	71	57	63	52
PSORT	53.2	74	51	62	72
TargetP	-	-	-	78	77
LOCnet	64.2	73	56	53	**86**
LOCSVMPSI*	73.2	79	69	85	64
PredLOC*	**78.1**	**80**	**74**	**87**	76

* NNPSL, SubLoc, LOCSVMPSI and PredLOC were trained using the same data set consisted of 2427 eukaryotic proteins. Results were obtained from Nair and Rost (2003) and Xie et al. (2005) for comparison with those of PredLOC.

Linux-compatible source codes and training files for PredLOC system are made available in www.ii.metu.edu.tr/~hogul/predLOC.htm, for non-commercial use of PredLOC system.

CHAPTER 5

Solvent Accessibility Prediction

Proteins are not alone when performing their biological activities. An important interaction occurs with the water molecules. Waters can touch residues at the surface of proteins. Each atom can potentially be touched by water, and the area of an atom on the surface that can be touched by water is called accessible surface area, or solvent-exposed area. The degree to which residues in the structure interact with solvent molecules, called *solvent accessibility*. Key functional properties of proteins and so-called active sites are strongly correlated with the solvent accessibility.

Solvent accessibility can be defined in two different ways. One is solvent accessibility percentage and it is defined as the ratio between the solvent accessible surface area of a residue and that in an extended tripeptide conformation. The second way is using relative categories in terms of a predefined threshold value of solvent accessibility. This categorization may be binary; buried or exposed, or ternary; buried, partially exposed, or exposed.

In this part of the thesis, the maximal unique match model that is introduced in Section 2.2.2 is evaluated such that it whether implies a conservation in residue interactions in protein sequence. Although solvent accessibility prediction is not a whole-sequence classification problem, rather it aims to classify a single residue inside the sequence, this problem is selected to discern the ability of maximal unique matches to detect the strict conservations over the sequence segments. Furthermore, residue feature representations are investigated and a slightly improved solution is presented.

5.1 Previous Studies

A few methods have been proposed for the prediction of solvent accessibility from the primary sequence of proteins (Chen, Zhou, Hu and Yoo, 2004). One group of methods is based on the single sequence prediction the solvent accessibility from local amino acid properties. Single sequence methods identify local statistics from amino acid sequences and predict the solvent accessibility using different classification schemes, such as Neural Networks (Rost and Sander, 1994; Ahmad and Gromiha, 2002; Ahmad, Gromiha, Sarai, 2003; Ahmad, Gromiha, Fawareh and Sarai, 2004), Bayesian statistics (Thomston and Goldstein, 1996) or Support Vector Machines (Yuan et al, 2002). The single sequence prediction accuracy is about 71% and this can be increased using multiple sequence information in the data set. Multiple sequence predictions use evolutionary information inferred from the profiles constructed by multiple sequence alignments (see Section 2.2.1). Multiple sequence methods increase the prediction accuracy up to about 79%. However, using multiple alignments is computationally inefficient and it is not always guaranteed that informative profiles are constructed in the given dataset.

5.2 Systems and Methods

A two-stage method is developed for the single sequence prediction of two-class (exposed or buried) solvent accessibility and ~73% accuracy is achieved with a 20% threshold in a nonredundant data set of 420 proteins. The first stage uses support vector machines to predict the two-class solvent accessibility using the residue features such as hydropathy scale and residue mass, as well as the neighborhood information from the left and right side of an amino acid. The second stage searches the maximal unique matches between the target protein and the data set. The SVM-based predictions are refined using the conservations over the maximal unique matches.

Baseline, SVM-based and MUM-based methods are evaluated for the prediction of solvent accessibility. Methods are applied individually and as an ensemble to test their performance over the protein data set.

5.2.1 Baseline Predictions

The baseline predictions can be obtained using the solvent accessibility statistics of each amino acid in the selected data set. Solvent accessibility values are taken from DSSP database (Kabsch and Sander, 1983) and the statistics are extracted from the training set. DSSP gives a solvent accessibility value between 0 and 9 for each residue of the proteins such that the value of 0 refers to a completely buried (0%) residue, 1 refers to a solvent accessibility of (0-11.1]%, 2 refers to (11.1-22.2]% and so on. The tendency of an amino acid to be buried is determined simply by comparing the counts of buried and exposed occurrences of that amino acid in the training set. If the number of buried occurrences is higher than exposed ones, this amino acid is predicted as buried for all test cases. Otherwise, it is marked as exposed. According to the statistics, V, I, L, F, M, W, C are buried and G, A, P, S, T, N, Q, Y, H, D, E, K, R are exposed amino acids with an accessibility threshold of 22.2%. For 0% threshold, all are marked as buried, whereas only G is exposed for 55.5% threshold.

5.2.2 SVM-based Prediction

To train the SVM classifier which makes a separation between buried and exposed classes (exposed used for positive and buried for negative classes), we used the feature vectors which represent the physicochemical properties of the amino acid, the amino acid itself and the properties of left and right neighbors of center amino acid to be predicted. In the previous work of Yuan et al (2002), only amino acid representations within a window are used for feature vectorization. The properties used in our vectorization step are listed in Table 17, where the hydropathy scales (free energy change for transfer from oil to water for each amino acids) are taken from the book of Horton, Moran, Ochs, Rawn and Scrimgeour (2002) and relative residue mass values are given by Li and Pan (2001). For each 3-length amino acid strings, a feature vector with a length of 66 is used. 20 of vector elements represent one of the 20 different amino acids and 2 elements are physicochemical properties explained before.

Table 17. Chemical and physical properties of amino acids used for feature vactorization

Amino acid	Hydropathy Scale	Relative residue mass (W as 1.0)
G	0.67	0.00076
A	1.0	0.115
V	2.3	0.33
I	3.1	0.13
L	2.2	0.13
F	2.5	0.7
P	-0.29	0.323
M	1.1	0.577
W	0.0	1.0
S	-1.1	0.238
T	-0.75	0.346
N	-2.7	0.446
Q	-2.9	0.55
Y	0.08	0.82
H	-1.7	0.63
D	-3.0	0.446
E	-2.6	0.55
K	-4.6	0.48
R	-7.5	0.777
C	0.17	0.36
X	0.0	0.5

5.2.3 MUM-based Refinement

In spite of the fact that the data set we used is composed of non-homolog or remote homolog proteins, they still share some conservative sub-patterns between them. If these kinds of sequence conservations refer also to the conservations in solvent accessible surfaces, we can use the statistics obtained from them to refine the incorrectly identified residues.

Since we have already trained 3-letter strings in SVM applications, here, we extract only the matches longer than 5 amino acids and calculate the solvent accessibility statistics obtained from the middle-point of each maximal unique match. Among all maximal unique matches extracted from the dataset, the percentage of correctly identified residues are 79.4%. For any homolog data set, this percentage promises good prediction accuracies for solvent accessibility. However, in our data set, containing less or no homology, the number of maximal unique matches is relatively low. Therefore, the information obtained from maximal unique matches can only be used for the refinement of the predictions made by other methods.

In MUM-based refinement stage, each protein in the test set is searched for maximal unique matches with all other proteins in a pairwise fashion. The solvent accessibility of a residue appearing in the middle of a maximal unique match is determined by a simple averaging scheme.

5.3 Experiments

The baseline predictions, SVM-based predictions, and MUM-based refinements are applied into data set for a solvent accessibility threshold of 22.2%. The data set contains 420 proteins which have no pair with a sequence similarity above 25%. In SVM-based prediction stage, 15 proteins which are randomly selected from the dataset are used for training the SVM. Total number of training examples is 3067, where 1564 of them are exposed and 1503 of them are buried with 22.2%

threshold. The remaining proteins are used for the tests. The proteins in the training and test sets are given in the Table 18 with their Protein Data Bank identification numbers.

All resulting predictions are compared with the actual values of solvent accessibilities obtained from DSSP database. The accuracy is defined as the percentage of number of correctly identified residues among all residues.

Table 18. Data set used in solvent accessibility prediction tests.

Training Set	1acx, 1amp, 1aya, 1ctf, 1hmp, 1hmy, 1hnf, 1hor, 5lyz, 6cpa, 6dfr, 6tmn, 7rsa, 9api, 9wga
Test Set	154l, 1aaz, 1add, 1ade, 1ahb, 1alk, 1amg, 1aor, 1aoz, 1asw, 1atp, 1avh, 1azu, 1bam, 1bbp, 1bcx, 1bdo, 1bds, 1bet, 1bfg, 1bmv, 1bnc, 1bov, 1bph, 1brs, 1bsd, 1cbg, 1cbh, 1cc5, 1cdl, 1cdt, 1cei, 1cel, 1cem, 1ceo, 1cew, 1cfb, 1cfr, 1cgu, 1chb, 1chd, 1chk, 1chm, 1cks, 1clc, 1cns, 1coi, 1col, 1com, 1cpc, 1cpn, 1cqa, 1crn, 1cse, 1csm, 1cth, 1ctn, 1ctm, 1ctn, 1ctu, 1cxs, 1cyx, 1daa, 1dar, 1del, 1dfj, 1dfn, 1dih, 1dik, 1din, 1dkz, 1dlc, 1dnp, 1dpg, 1dsb, 1dts, 1dup, 1dyn, 1eca, 1ece, 1ecl, 1ecp, 1edd, 1edm, 1edn, 1eft, 1efu, 1epb, 1ese, 1esl, 1etu, 1euu, 1fba, 1fbl, 1fc2, 1fdl, 1fdt, 1fdx, 1fin, 1fjm, 1fkf, 1fnd, 1fua, 1fuq, 1fxi, 1gal, 1gcb, 1gcm, 1gd1, 1gdj, 1gep, 1gfl, 1ghs, 1gky, 1gln, 1gmp, 1gnd, 1gog, 1gp1, 1gp2, 1gpc, 1gpm, 1grj, 1gtm, 1gtq, 1gym, 1han, 1hip, 1hcg, 1hcr, 1hiw, 1hjr, 1hpl, 1hsl, 1htr, 1hup, 1hvq, 1hxn, 1hyp, 1il8, 1ilk, 1inp, 1irk, 1isa, 1isu, 1jud, 1kin, 1knb, 1kpt, 1krc, 1kte, 1ktq, 1kuh, 1l58, 1lap, 1lat, 1lba, 1lbu, 1leh, 1lib, 1lis, 1lki, 1lpb, 1lpe, 1mai, 1mas, 1mct, 1mda, 1mdt, 1mjc, 1mla, 1mmo, 1mns, 1mof, 1mrr, 1mrt, 1msp, 1nal, 1nar, 1nba, 1ncg, 1ndh, 1nfp, 1nga, 1nlk, 1nol, 1nox, 1noz, 1oac, 1onr, 1otg, 1ovb, 1ovo, 1oxy, 1oyc, 1paz, 1pbp, 1pbw, 1pda, 1pdn, 1pdo, 1pga, 1pht, 1pii, 1pky, 1pmi, 1pnm, 1pnt, 1poc, 1pow, 1ppi, 1ppt, 1ptr, 1ptx, 1pyp, 1pyt, 1qbb, 1qrd, 1r09, 1rbp, 1rec, 1reg, 1req, 1rhd, 1rhg, 1rie, 1ris, 1rld, 1rlr, 1rpo, 1rsy, 1rvv, 1s01, 1scu, 1sei, 1ses, 1sfe, 1sft, 1sh1, 1smn, 1smp, 1spb, 1sra, 1srj, 1stf, 1stm, 1svb, 1tab, 1taq, 1tcb, 1tcr, 1tfr, 1tht, 1thx, 1tie, 1tif, 1tig, 1tii, 1tml, 1tnd, 1tnf, 1tpl, 1trb, 1trh, 1trk, 1tsp, 1tss, 1tul, 1tup, 1ubd, 1ubq, 1udh, 1umu, 1vca, 1vcc, 1vhh, 1vhr, 1vid, 1vjs, 1vmo, 1vnc, 1vok, 1vpt, 1wap, 1wfb, 1whi, 1wsy, 1xva, 1ypt, 1yrn, 1znb, 1zym, 256b, 2aai, 2aat, 2abk, 2adm, 2afn, 2ak3, 2alp, 2asr, 2bat, 2blt, 2bop, 2cab, 2ccy, 2cmd, 2cpo, 2cyp, 2dkb, 2dln, 2dnj, 2ebn, 2end, 2erl, 2fox, 2fxb, 2gbp, 2gcr, 2gls, 2gn5, 2gsq, 2hft, 2hhm, 2hip, 2hmz, 2hpr, 2i1b, 2ltn, 2mev, 2mhu, 2mlt, 2mta, 2nad, 2npx, 2olb, 2pab, 2pgd, 2phh, 2phy, 2pol, 2reb, 2rsl, 2rsp, 2scp, 2sil, 2sns, 2sod, 2spt, 2stv, 2tgi, 2tgp, 2tmd, 2tmv, 2trt, 2tsc, 2utg, 2wrp, 2yhx, 3ait, 3b5c, 3bcl, 3blm, 3cd4, 3chy, 3cla, 3cln, 3cox, 3eca, 3gap, 3hmg, 3icb, 3ink, 3mdd, 3pgk, 3pgm, 3pmg, 3rnt, 3tim, 4bp2, 4cpa, 4fis, 4gr1, 4pfk, 4rhv, 4rxn, 4sdh, 4sgb, 4ts1, 4xia, 5cyt, 5er2, 5ldh, 5sic, 6acn, 6cpp, 6cts, 6hir, 6rlx, 7cat, 7icd, 821p, 8adh, 9ins, 9pap

5.4 Results and Discussion

The experimental results are given in Table 19 with varying threshold values. As we can see from the table, SVM-based predictions give an improvement of 2.8%

over baseline predictions for a 22.2% threshold, which is the case of evenly distribution of buried and exposed classes. When applied on the baseline predictions, MUM-based refinement improves the baseline accuracy by 0.5%. The MUM-based refinement improves the SVM-based predictions by 0.4%. Overall improvement achieved by the combination of SVM-based and MUM-based predictions over the baseline accuracy is 3.2%. The methods are also tested for 0% and 55.5% thresholds and the results are given in the Table 19. For those threshold values, same improvement can not be achieved with SVM. This is probably due to the fact that the positive and negative examples are not evenly distributed for those threshold values.

Table 19. Results showing the accuracies achieved with different solvent accessibility thresholds

Threshold	**0%**	**22.2%**	**55.5%**
Method			
Baseline	75.3%	69.5%	79.6%
Previous SVM method (Yuan et al, 2002)	70.9%	71.4%	78.7%
New SVM method with extended features	71.6%	72.3%	79.1%
MUM-refinement over baseline	75.7%	70.0%	79.8%
MUM-refinement over new SVM	72.3%	72.7%	79.3%

Since there is no common benchmarking set for the solvent accessibility prediction, a direct comparison with the previous methods that used different data sets is not valid. According to the recent review of Chen et al. (2004), which reports a baseline prediction accuracy of 69.6% in their data set, the accuracies achieved with the tested methods are 71.5% for decision tree model and 71.2% for Bayesian statistics with a 20% threshold. We could make a fair comparison only with the baseline method and the previous SVM method of Yuan et al. (2002) with the same threshold over the same experimental setup (Table 19). Comparing with these results, new methods achieve slightly better accuracy.

CHAPTER 6

Conclusions and Final Remarks

Throughout the thesis, three novel methods are introduced for computational representation of protein sequences and their applicability are discussed as a result of various practical applications that are severely important in computational biology. In this chapter, the conclusions inferred from the conducted research are presented together with the recommendations related to further studies.

Composition gives important clues. Amino acid composition has been surprisingly effective in structural classification, however it has failed due the lack of local order information when the sequences become more divergent. Motivating from the fact that dipeptide composition has provided a greater accuracy, longer *n*-peptides are employed in this study over a common SVM-based classification framework. This scheme is shown to provide a significant improvement in the prediction accuracy comparing with amino acid and dipeptide compositions in both experiments conducted for remote homology detection and subcellular localization prediction. Moreover, the homology detection accuracy achieved by *n*-peptide composition is comparable to that of the feature representations based on pairwise similarity scores, whereas it provides much more efficient solution due to the reasons explained below.

Use of reduced amino acid alphabets has many advantages. Main disadvantage of using *n*-peptide compositions, which is possibly the reason for the non-existence of any previous attempt to use them, is the time and space complexity of extraction and management of corresponding feature vectors. The proposal of this thesis on the gradual reduction of amino acid alphabets for larger values of *n* provides a complete solution to these problems. With this solution, the feature space becomes linearly dependent on the length of the *n*-peptides to be searched. Not only providing an efficient space complexity, this scheme also allows the evaluation of possible mismatches in longer *n*-peptides, which is a natural case in the evolution of proteins. The possibility of implementing *n*-peptide searches with a hash structure also reduces the time complexity of the feature vector construction.

Maximal unique match approach is simple but successful for measuring distant similarities. This thesis is the first time that the maximal unique match definition is applied for protein sequence similarity scoring. In spite of their simplicity and efficiency, the MUM-based similarity scores provided better accuracy than most widely used BLAST scores over the discriminative homology detection framework and performed nearly as well as the dynamic-programming-based alignment scores. When used for the refinement of residue solvent accessibility predictions, MUMs became successful to identify most conserved residues although their use did not make a significant improvement in resulting prediction accuracy due to the small and non-redundant dataset. The refinement methodology promises better improvements in protein residue classification when larger datasets are made available.

Pairwise probabilistic suffix trees are much more efficient yet accurate. Another sequence comparison method introduced in the thesis is based on the probabilistic suffix trees. To our knowledge, this is the first application of PSTs for pairwise sequence comparison. A number of modifications are described in the tree construction and prediction phases of original family-based PST method so that it can be adopted for detecting distant pairwise similarities. When combined with SVMs, pairwise PST scores provided greater accuracy for detecting remote homologies in comparison with the family-based PST model. The same representation was used in subcellular localization prediction tests, and became the core part of the new prediction tool, called PredLOC.

Homology modeling will probably remain central for long years. Owing to many unsuccessful attempts for *ab inito* prediction of protein structure, sequence-based homology modeling via sequence similarity is still considered to be the best solution for protein structure annotations. Although the system fails when no homolog is found, a significant improvement has been achieved in the detection of distant relationships in recent years. This study makes a valuable contribution to the studies on remote homology detection. While most accurate methods suffer from the computational inefficiency, which makes the methods impractical to use, the new methods introduced in this thesis provide considerably improved efficiency with preserved level of accuracy as the state-of-the-art methods have achieved. With these significant progress either made by this study or other researches, it is expected that homology modeling will continue to be used for many years as long as the real biochemical and biophysical processes behind protein folding and functioning remain unsolved.

Many thing effects subcellular localization. Varying techniques and ideas, such as targeting signal detection, homology modeling, amino acid and dipeptide compositions with machine learning algorithms, were used for the prediction of subcellular localization of proteins in the past years. Some of the studies attempted to integrate multiple modules that use different approaches to increase the accuracy of the predictions. In this study, a new method, called as PredLOC, is introduced and its prediction accuracy is reported on two independent test sets. The new method combines, either in an implicit or explicit way, many of the ideas used in the previous works into a single module. First, homology information is used by realizing pairwise sequence comparisons in given data set. These pairwise similarity scores are combined into feature vectors to train SVM, which is a common machine learning technique used by many of the recent and relatively successful systems. For pairwise sequence comparison, a new technique based on

PSTs is used. In contrast to sequence alignment, the PST-based comparison does not care of the order of residue or segment matches in the sequences being compared. This provides the opportunity of incorporating the effect of some problem specific motifs into the prediction model by rewarding the significant matches between the sequences. Targeting signals generally occurring in N-terminal sequences are also taken into account by integrating N-terminal similarity scores into feature vectors. The use of reduced alphabets in *n*-peptide compositions for larger values of *n* provides the opportunity to evaluate the longer local amino acid ordering properties with evolutionarily possible modifications, which are not taken into account in the PST-based comparison. According to the experimental results, PredLOC significantly outperforms many of the existing methods and apparently becomes a powerful alternative to available subcellular localization predictors. All previous studies, in consistent with the results of this thesis, suggest that many parameters take role in subcellular localization of proteins. In the light of those discussions, it is believed that the protein encoding schemes presented in this thesis can also be combined with other schemes for further improvements in the prediction accuracy.

Solvent accessibility prediction is a difficult problem. Protein solvent accessibility is an important property for the annotation of newly extracted protein sequences. A new computational method is introduced for the prediction of solvent accessibility using solely the sequence information and the results of the tests performed on a non-redundant protein set are presented. The new method uses an improved SVM approach with extended features for the prediction of the accessibilities and refines the SVM predictions along with the pairwise conservations, i.e. maximal unique matches, between the sequences. The main reason for the improvement in SVM predictions is the incorporation of new physicochemical features of the protein residues in the vectorization phase. Although the maximal unique match refinement does not make a significant improvement on the accuracy, it promises good results when the sufficient number of homologs is found. On the other hand, we could not witness a significant improvement on the prediction of solvent accessibility in the last decade. This disappointment suggests that the solvent accessibility is a hard problem in that only local sequence information can not provide sufficient knowledge. Indeed, it is currently not clear that what other properties of proteins play role in solvent accessibility.

New methods will find applications in many other problems. Novel sequence representations introduced in this thesis were employed for different biological problems and they are shown to be valuable especially in sequence-based protein classification problems. Moreover, the new methods are believed to find applications in many other problems related with biomolecular sequences.

There are many categorization problems in which the proteins are required to be assigned to one of the well-known classes; recognizing protein folds (Wallqvist, Fukunishi, Murphy, Fadel and Levy, 2002; Schonbrun, Wedemeyer and Baker, 2002; Ding and Dubchak, 2001), identifying functional categories (Cai et al., 2003; Cathy, Wu, Huang, Yeh and Barker, 2003), predicting enzyme classes (Ben-hur and Brutlag, 2003), detecting membrane and outer-membrane proteins (Gromiha and Suwa, 2005), predicting biological process that the protein is involved etc. Although two protein classification problems were examined in this thesis for the applicability of the new sequence representation schemes, the methods can simply be adopted for all other problems exemplified above.

In spite of the fact that the PST-based sequence comparison method is devised to be adopted in the similarity-score-based sequence representation scheme, it is applicable to other problems regarding sequence analysis. The pairwise PST model may provide a detailed view of highly correlated segments between the sequences. Therefore, it can be used for detecting subtle motifs that may be evolutionarily related. While the sequence alignment methods attempt to keep the order of residues unchanged, the new method does not consider the global positions of amino acids in the sequences. This fact brings up with a potential ability of the new scheme to identify the possible rearrangement mutations, which cannot be detected by an alignment. Not only for proteins, the methods can also be used for the problems related to DNA and RNA sequences by only replacing the amino acid alphabets with the corresponding 4-letter alphabets.

Cooperation between computer scientists and bioscientists is severely important. Bioinformatics is an emerging field and becoming increasingly important with the continuous accumulation of genetics and proteomics data. Drug discovery is expected to be one of the central problem on which the scientists will focus over the next century. This and related problems will require an extensive support of computer and information sciences. In this respect, it can not be thought that computer scientists and bioscientists work independently to make a significant progress in the field. Higher synergy between them will be resulted in the inventions that are more beneficial to human life.

REFERENCES

Ahmad S., Gromiha M, Fawareh H., Sarai A. (2004). ASAView: Database and tool for solvent accessibility representation in proteins. BMC Bioinformatics, 5, 51-56.

Ahmad S., Gromiha M., Sarai A. (2003). Real value prediction of solvent accessibility from amino acid sequence. Proteins: Structure, Function, and Genetics, 50, 629-635.

Ahmad S., Gromiha M.M. (2002). NETASA: neural network based prediction of solvent accessibility. Bioinformatics, 18, 819-824.

Altschul S., Gish W., Miller W., Myers E. W., Lipman D. (1990). A basic local alignment search tool. Journal of Molecular Biology, 251, 403-10.

Altschul S., Madden T., Schaffer A., Zhang J., Zhang Z., Miller W., Lipman D. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research, 25, 3389-3402.

Anfinsen C. B. (1973). Studies on the principles that govern the folding of protein chains. Science, 181, 223-230.

Apostolico A., and Bejerano G. (2000). Optimal amnestic probabilistic automata or how to learn and classify proteins in linear time and space. Journal of Computational Biology, 7, 381-393.

Atalay V., Atalay R.Ç. (2005) Implicit motif distribution based hybrid computational kernel for sequence classification. Bioinformatics, 21, 1429-1436.

Bahar I., Atilgan A.R., Jernigan R.J., Erman B. (1997). Understanding the recognition of protein structural classes by amino acid composition. Proteins, 29, 172-185.

Bairoch A. and Apweiler R. (1999). The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1999. Nucleic Acids Research, 27, 49-54.

Bateman A., Birney E., Cerruti L., Durbin R., Etwiller L., Eddy S. R., Griffiths-Jones S., Howe K. L., Marshall M., Sonnhammer E. L. (2002). The Pfam Protein Families Database. Nucleic Acids Research, 30, 276-280

Bejerano G. (2004). Algorithms for variable length Markov chain modeling. Bioinformatics, 20, 788-789.

Bejerano G., Yona G. (2000). Variations on probabilistic suffix trees: statistical modeling and prediction of protein families. Bioinformatics, 17, 23-43.

Ben-hur A., Brutlag D. (2003). Remote homology detection: a motif based approach. Bioinformatics, 19, 26-33.

Berman H. M., Westbrook J., Feng Z., Gilliland G., Bhat T. N., Weissig H., Shindyalov I. N., Bourne P. E. (2000). The Protein Data Bank. Nucleic Acids Research, 28, 235-242.

Bhasin M., Garg A., Raghava G.P.S. (2005). PSLpred: Prediction of subcellular localization of bacterial proteins. Bioinformatics, 21, 2522-2524.

Bhasin M. and Raghava G.P.S. (2004) ESLpred: SVM-based method for subcellular localization of eukaryotic proteins using dipeptide composition and PSI-BLAST. Nucleic Acids Research, 32, 415–419

Bieganski P., Riedl J., Carlis J. V., Retzael E. R. (1994). Generalized suffix trees for biological sequence data: applications and implementations. Proceeding of the 27th Hawaii International Conference on Systems and Sciences, 35-44.

Bock J. R., Gough D. A. (2001). Predicting protein-protein interactions from primary structure. Bioinformatics, 17, 455-460.

Bystroff C., Baker D. (1998). Prediction of local structure in proteins using a library of sequence-structure motifs. Journal of Molecular Biology, 281, 565-577.

Cai C. Z., Wang W. L., Sun L. Z., Chen Y. Z. (2003). Protein function classification via support vector machines. Mathematical Biosciences, 185, 111-122.

Carillo H., Lipman D. J. (1988). The multiple alignment problem in biology. SIAM Journal of Applied Mathematics, 48, 1073-1082.

Cathy H. Wu C.H., Huang H., Yeh L.L., Barker W.C. (2003). Protein family classification and functional annotation. Computational Biology and Chemistry, 27, 37-47.

Chen H., Zhou H., Hu X., Yoo I. (2004) Classification comparison of prediction of solvent accessibility from protein sequences. 2nd Asia-Pacific Bioinformatics Conference, Dunedin, New Zelland.

Cokol M., Nair R., Rost B. (2000). Finding nuclear localisation signals. EMBO Reports, 1, 411-415.

Dayhoff M. O., Schwartz R. M., Orcutt B. C. (1978). A model of evolutionary change in proteins. Atlas of Protein Sequence and Structure, 5, 345-52.

Delcher A., Kasif S., Fleishmann R., Peterson J., White O., Salzberg S. (1999). Alignment of whole genomes. Nucleic Acids Research, 27, 2369-2376.

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

Eddy S. R. (1998). Profile hidden Markov models, Bioinformatics, 14, 755, 763.

Eisenhaber F., Eisenhaber B., Kubina W., Maurer-Stroh S., Neuberger G., Schneider G., Wildpaner M. (2003). Prediction of lipid posttranslational modifications and localization signals from protein sequences: big-Pi, NMT and PTS1. Nucleic Acids Research, 31, 3631-3634.

Emanuelsson O., Nielsen H., Brunak S., Gunnar H. (2000). Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. Journal of Molecular Biology, 300, 1005-1016.

Falquet L., Pagni M., Bucher P., Hulo N., Sigrist C., Hofmann K., Bairoch A. (2002). The PROSITE database, its status in 2002. Nucleic Acid Research, 30, 235-238.

Gardy J.L., Laird M.R. , Chen F., Rey S., Walsh C.J., Ester M., Brinkman F.S.L. (2005) PSORTb v.2.0: expanded prediction of bacterial protein subcellular localization and insights gained from comparative proteome analysis. Bioinformatics, 21, 617–623.

Gribskov M., Lüthy R., Eisenberg D. (1990). Profile Analysis. Methods in Enzymology, 183, 146-59.

Gribskov, M., and Robinson, N.L. (1996). Use of receiver operating characteristic analysis to evaluate sequence matching. Computers and Chemistry, 20, 25–33.

Gromiha M.M, Suwa M. (2005). A simple statistical method for discriminating outer membrane proteins with better accuracy. Bioinformatics, 21, 961-968.

Grundy W. N. (1998). Homology Detection via Family Pairwise Search. Journal of Computational Biology, 5, 479-492.

Gusfield D. (1997). Algorithms on Strings, Trees, and Sequences: Computer science and Computational Biology, Cambridge University Press, New York.

Henikoff S. and Henikoff J. G., 1992, Amino acid substitution matrices from protein blocks. Proc. Natl. Acad. Sci.USA, 89, 10915-10919.

Henikoff S., Henikoff J., Pietrokovski S. (1999). BLOCKS: a nonredundant database of protein alignment blocks derived from multiple compilations. Bioinformatics, 15, 471-479.

Higgins D. G., Thompson J. D., Gibson T. J. (1994). CLUSTAL W: Improving the sensitivity of progressive multiple alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acid Research, 22 ,4673-80.

Hirschberg D. S. (1977). Algorithms for the longest common subsequence problem. Journal of ACM, 24, 664-75.

Horton H.B., Moran L.A., Ochs R.S., Rawn J.D., Scrimgeour K.G. (2002). Principles of Biochemistry, Prentice Hall.

Horton P., Nakai K. (1999). PSORT: a program for detecting sorting signals in proteins and predicting their subcellular localization. Trends in Biochemical Sciences, 24, 34–36.

Hou Y., Hsu W., Lee M.L., Bystroff C. (2003). Efficient remote homology detection using local structure. Bioinformatics, 19, 2294-2301.

Hua, S. and Sun, Z. (2001). Support vector machine approach for protein subcellular localization prediction. Bioinformatics, 17, 721–728.

Huang J. and Brutlag D. (2001). The emotif database. Nucleic Acid Research, 29, 202-204.

Huang Y., Li Y. (2004). Prediction of protein subcellular locations using fuzzy k-NN method. Bioinformatics, 20, 21-28.

Jaakola T., Diekhans M., Haussler D. (2000). A discriminative framework for detecting remote protein homologies. Journal of Computational Biology, 7, 95-114.

Jin L., Weiwu F., Tang H. (2003). Prediction of protein structural classes by a new measure of information discrepancy. Computational Biology and Chemistry, 27, 373-380.

Kabsch W., Sander C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen bonded and geometrical features. Biopolymers, 22, 2577-637.

Karchin R., Karplus K., Haussler D. (2002). Classifying G-protein coupled receptors with support vector machines. Bioinformatics, 18, 147-159.

Karplus K., Barrett C., Hughey R. (1998). Hidden Markov Models for detecting remote protein homologies. Bioinformatics, 14, 846-856.

Kim J., Pramanik S., Chung M. J. (1994). Multiple sequence alignment using simulated annealing. Computer Applications in Biosciences, 10, 419-426.

Krogh A., Brown M., Mian I.S, Sjölander K., Haussler D. (1994). Hidden Markov models in computational biology: applications to protein modeling. Journal of Molecular Biology, 235, 1501-1531.

Lawrence C. E., Altschul S. F., Boguski M. S., Liu J. S., Neuwald A. F., Wootton J. C. (1993). Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. Science, 262, 208-214.

Li X, Pan X-M. (2001). New method for accurate prediction of solvent accessibility from protein sequence. Proteins: Structure, Function, and Genetics, 42, 1-5.

Li T., Fan K., Wang J., Wang W. (2003). Reduction of protein sequence complexity by residue grouping. Protein Engineering, 16, 323-330.

Liao L., Noble W. S. (2003). Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. Journal of Computational Biology, 10, 857-868.

Lipman D. J., Pearson, W. R. (1985). Rapid and sensitive protein similarity search. Science, 227, 1435-41.

Liu X., Liu D., Qi J., Zheng W. (2002). Simplified amino acid alphabets based on deviation of conditional probability from random background. Physical Reviews E, 66, 021960.

Löytynoja A., Milinkovitch M. C. (2003). A hidden Markov Model for progressive multiple alignment. Bioinformatics, 19, 1505-1513.

Matthews B.W. (1975). Comparison of predicted and observed secondary structure of T4 phage lysozyme. Biochim. Biophys. Acta, 405, 442–451.

McCreight E. M. (1976). A space-economical suffix tree construction algorithm. Journal of ACM, 23, 262-72.

Murphy L.R., Wallqvist A., Levy R.M. (2000). Simplified amino acid alphabets for protein fold recognition and implications for folding. Protein Engineering, 13, 149-152.

Murzin A. G., Brenner S. E., Hubbard T., Chothia C (1995). SCOP: A structural classification of proteins database for the investigation of sequences and structures. Journal of Molecular Biology, 247, 536-40.

Myers E. W. (1991). An overview of sequence comparison algorithms in molecular biology. Technical Report, TR 91-29, University of Arizona.

Myers E. W., Miller W. (1988). Optimal alignments in linear space. Computer Applications in Biosciences, 4, 11-17.

Nair R., Rost B. (2003). Better prediction of sub-cellular localization by combining evolutionary and structural information. Proteins: Structure, Function and Genetics, 53, 917-930.

Nakai K. and Brinkman S.L. (2003). PSORT-B: improving protein subcellular localization prediction for Gram-negative bacteria. Nucleic Acids Research, 31, 3613-3617.

Nakashima H., Nishikawa K. (1994). Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies. Journal of Molecular Biology, 238, 54-61.

Needleman S. B., Wunch C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology, 48, 443-53.

Nielsen H., Engelbrecht J., Brunak S. , von Heijne G. (1997). Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. Protein Engineering, 10, 1-6.

Notredame C., Higgins D. G. (1996). SAGA: Sequence alignment by genetic algorithms. Nucleic Acids Research, 24, 1515-1524.

Oğul H., Erciyes K. (2001). Identifying all local and global alignments between two DNA sequences. Proc. 17th Int. Sym. on Computer and Information Sciences, 468-475.

Orengo C.A., Michie A.D., Jones S., Jones D.T., Swindells M.B., and Thornton J.M. (1997). CATH: A Hierarchic Classification of Protein Domain Structures. Structure, 5, 1093-1108.

Park J., Karplus K., Barrett C., Hughey R., Haussler D., Hubbard T., Chothia C. (1998). Sequence comparisons using multiple sequences detect tree times as many remote homologues as pairwise methods. Journal of Molecular Biology, 284, 1201-1210.

Park K.J., Kanehisa M. (2003). Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. Bioinformatics, 19, 1656-1563.

Reinhardt A., Hubbard T. (1998). Using neural networks for prediction of the subcellular location of proteins. Nucleic Acids Research, 26, 2230–2236.

Richardson C.J., Barlow D.J. (1999). The bottom line for prediction of residue solvent accessibility. Protein Engineering, 12, 1051-1054.

Rost B., Sander C. (1994). Conservation and prediction of solvent accessibility in protein families. Proteins: Structure, Function, and Genetics, 20, 216-226.

Rost B. (1999). Twilight zone of protein sequence alignments. Protein engineering, 12, 85-94.

Schonbrun J., Wedemeyer W. J., Baker D. (2002). Protein structure prediction in 2002. Current Opinion in Structural Biology, 12, 348-354.

Smith T. F., Waterman M. S. (1981). Identification of common molecular subsequences. Journal of Molecular Biology, 147, 195-97.

Thompson M. J., Goldstein R. A. (1996). predicting solvent accessibility: higher accuracy using Bayesian statistics and optimized residue substitution classes. Proteins: Structure, Function, and Genetics, 25, 38-47.

Ukkonen E. (1995). On-line construction of suffix-trees. Algorithmica, 14, 249-60.

Vapnik V. (1995). The nature of statistical learning theory, Springer-Verlag, New York.

Vapnik V., Cortes C. (1995). Support vector networks. Machine Learning, 20, 73-93

Wallqvist A., Fukunishi Y., Murphy L. R., Fadel A., Levy R. M. (2000). Iterative sequence/structure search for protein homologs: comparison with amino acid sequence alignments and application to fold recognition in genome databases. Bioinformatics, 16, 988-1002.

Ward J., McGuffin L. C., Buxton B. F., Jones D. T. (2003). Secondary structure prediction with support vector machines. Bioinformatics, 19, 1650-55.

Weiner P. (1977). Linear pattern matching algorithms, Proceeding of the 14th IEEE Symposium on Switching and Automata Theory, 1-11.

Weis K. (1998). Importins and exportins: how to get in and out of the nucleus. Trends in Biochemical Sciences, 67, 265-306.

Xie D.,Li A.,Wang M., Fan Z., Feng H. (2005). LOCSVMPSI: a web server for subcellular localization of eukaryotic proteins using SVM and profile of PSI-BLAST. Nucleic Acids Research, 33, W105-W110.

Yu C., Lin C., Hwang J. (2004). Predicting subcellular localization of proteins for Gram-negative bacteria by support vector machines based on n-peptide compositions. Protein Science, 13, 1402–1406

Yuan Z., Burrage K., Mattick J. (2002). Prediction of protein solvent accessibility using support vector machines. Proteins: Structure, Function, and Genetics, 48, 566-570.

Zhang C.T., Chou K.C., Maggiora G.M. (1995). Predicting protein structural classes from amino acid composition: Application of fuzzy clustering. Protein Engineering, 8, 425–435.

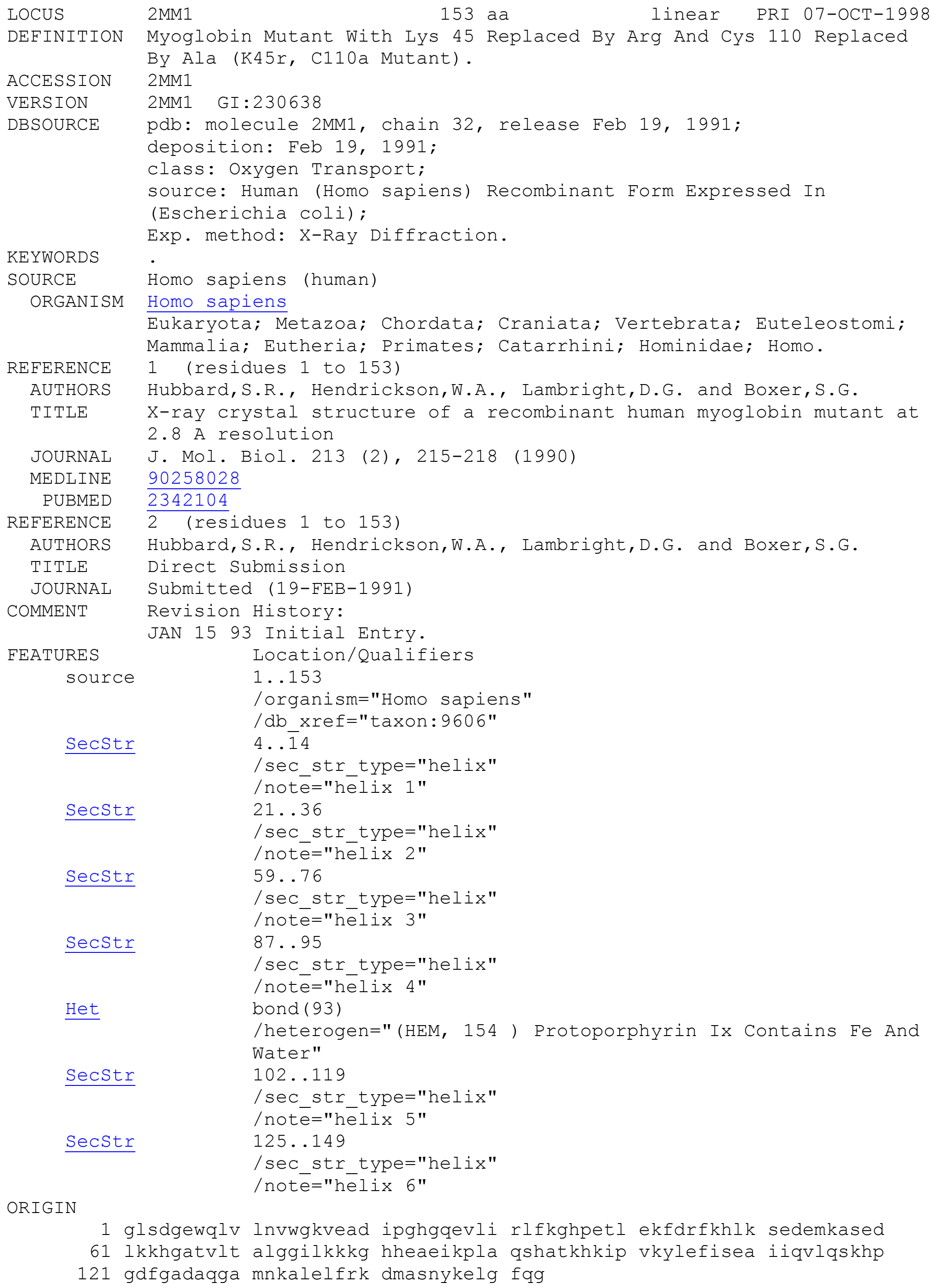
APPENDICES

APPENDIX A. NCBI

National Center for Biotechnology Information provides an integrated tool for all kind of biological information. There are several databases and search tools in the online NCBI web page. The tools in NCBI include:

- Literature database; an extended searchable literature library of life sciences,
- Entrez database; a retrieval system designed for searching several linked databases including PDB, SwissProt, PIR,
- Nucleotide database; accepted genome data from sequencing projects from around the world,
- Genome-specific resources,
- Tools for data mining,
- Tools for sequence analysis, e.g. BLAST searches all databases for similar sequences for a given sequence.
- Tools for 3D structure display and similarity searching

The NCBI home page is http://www.ncbi.nlm.nih.gov. An example search result of NCBI for protein 2mm1 is given in Figure 29.

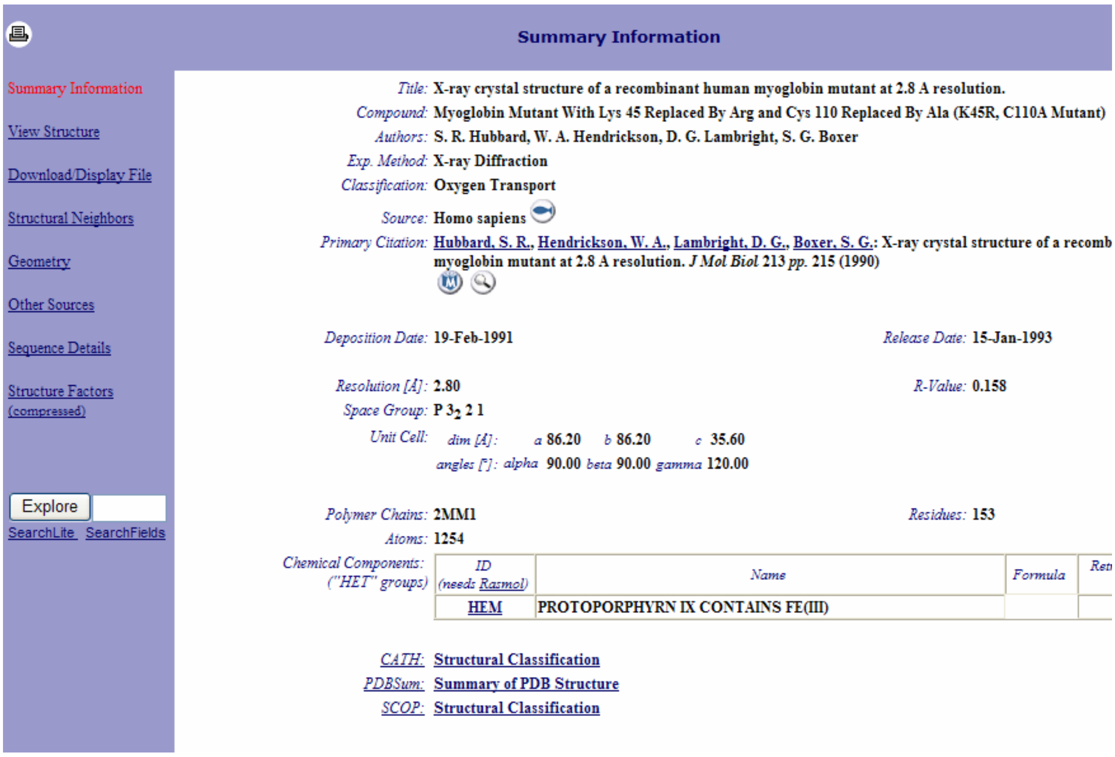


```
LOCUS       2MM1                     153 aa            linear   PRI 07-OCT-1998
DEFINITION  Myoglobin Mutant With Lys 45 Replaced By Arg And Cys 110 Replaced
            By Ala (K45r, C110a Mutant).
ACCESSION   2MM1
VERSION     2MM1  GI:230638
DBSOURCE    pdb: molecule 2MM1, chain 32, release Feb 19, 1991;
            deposition: Feb 19, 1991;
            class: Oxygen Transport;
            source: Human (Homo sapiens) Recombinant Form Expressed In
            (Escherichia coli);
            Exp. method: X-Ray Diffraction.
KEYWORDS    .
SOURCE      Homo sapiens (human)
  ORGANISM  Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo.
REFERENCE   1  (residues 1 to 153)
  AUTHORS   Hubbard,S.R., Hendrickson,W.A., Lambright,D.G. and Boxer,S.G.
  TITLE     X-ray crystal structure of a recombinant human myoglobin mutant at
            2.8 A resolution
  JOURNAL   J. Mol. Biol. 213 (2), 215-218 (1990)
  MEDLINE   90258028
   PUBMED   2342104
REFERENCE   2  (residues 1 to 153)
  AUTHORS   Hubbard,S.R., Hendrickson,W.A., Lambright,D.G. and Boxer,S.G.
  TITLE     Direct Submission
  JOURNAL   Submitted (19-FEB-1991)
COMMENT     Revision History:
            JAN 15 93 Initial Entry.
FEATURES             Location/Qualifiers
     source          1..153
                     /organism="Homo sapiens"
                     /db_xref="taxon:9606"
     SecStr          4..14
                     /sec_str_type="helix"
                     /note="helix 1"
     SecStr          21..36
                     /sec_str_type="helix"
                     /note="helix 2"
     SecStr          59..76
                     /sec_str_type="helix"
                     /note="helix 3"
     SecStr          87..95
                     /sec_str_type="helix"
                     /note="helix 4"
     Het             bond(93)
                     /heterogen="(HEM, 154 ) Protoporphyrin Ix Contains Fe And
                     Water"
     SecStr          102..119
                     /sec_str_type="helix"
                     /note="helix 5"
     SecStr          125..149
                     /sec_str_type="helix"
                     /note="helix 6"
ORIGIN
        1 glsdgewqlv lnvwgkvead ipghgqevli rlfkghpetl ekfdrfkhlk sedemkased
       61 lkkhgatvlt alggilkkkg hheaeikpla qshatkhkip vkylefisea iiqvlqskhp
      121 gdfgadaqga mnkalelfrk dmasnykelg fqg
```

Figure 29. NCBI search result for 2mm1.

APPENDIX B. PDB

Protein Data Bank (Berman et al., 2000) is one of the largest protein database in the world. A primary key, named as PDB ID, is maintained for each protein in the database, which unifies the protein entries. The database contains many information about the proteins; primary sequence, tertiary structure, i.e. all atomic positions of proteins in *xyz* coordinate space, source of organism, deposition and release dates, compounds, classifications and authors, links to other databases, etc. A sample view of PDB entry is given in Figure 30. The database is available online in http://www.rcsb.org/pdb/.



Summary Information

Title: X-ray crystal structure of a recombinant human myoglobin mutant at 2.8 A resolution.
Compound: Myoglobin Mutant With Lys 45 Replaced By Arg and Cys 110 Replaced By Ala (K45R, C110A Mutant)
Authors: S. R. Hubbard, W. A. Hendrickson, D. G. Lambright, S. G. Boxer
Exp. Method: X-ray Diffraction
Classification: Oxygen Transport
Source: Homo sapiens
Primary Citation: Hubbard, S. R., Hendrickson, W. A., Lambright, D. G., Boxer, S. G.: X-ray crystal structure of a recomb myoglobin mutant at 2.8 A resolution. *J Mol Biol* 213 *pp.* 215 (1990)

Deposition Date: 19-Feb-1991 *Release Date:* 15-Jan-1993

Resolution [Å]: 2.80 *R-Value:* 0.158
Space Group: P 3_2 2 1
Unit Cell: *dim [Å]:* *a* 86.20 *b* 86.20 *c* 35.60
angles [°]: *alpha* 90.00 *beta* 90.00 *gamma* 120.00

Polymer Chains: 2MM1 *Residues:* 153
Atoms: 1254
Chemical Components: *("HET" groups)*

ID *(needs Rasmol)*	*Name*	*Formula*	*Ret*
HEM	PROTOPORPHYRN IX CONTAINS FE(III)		

CATH: Structural Classification
PDBSum: Summary of PDB Structure
SCOP: Structural Classification

Figure 30. PDB search result for 2mm1.

APPENDIX C. SWISSPROT

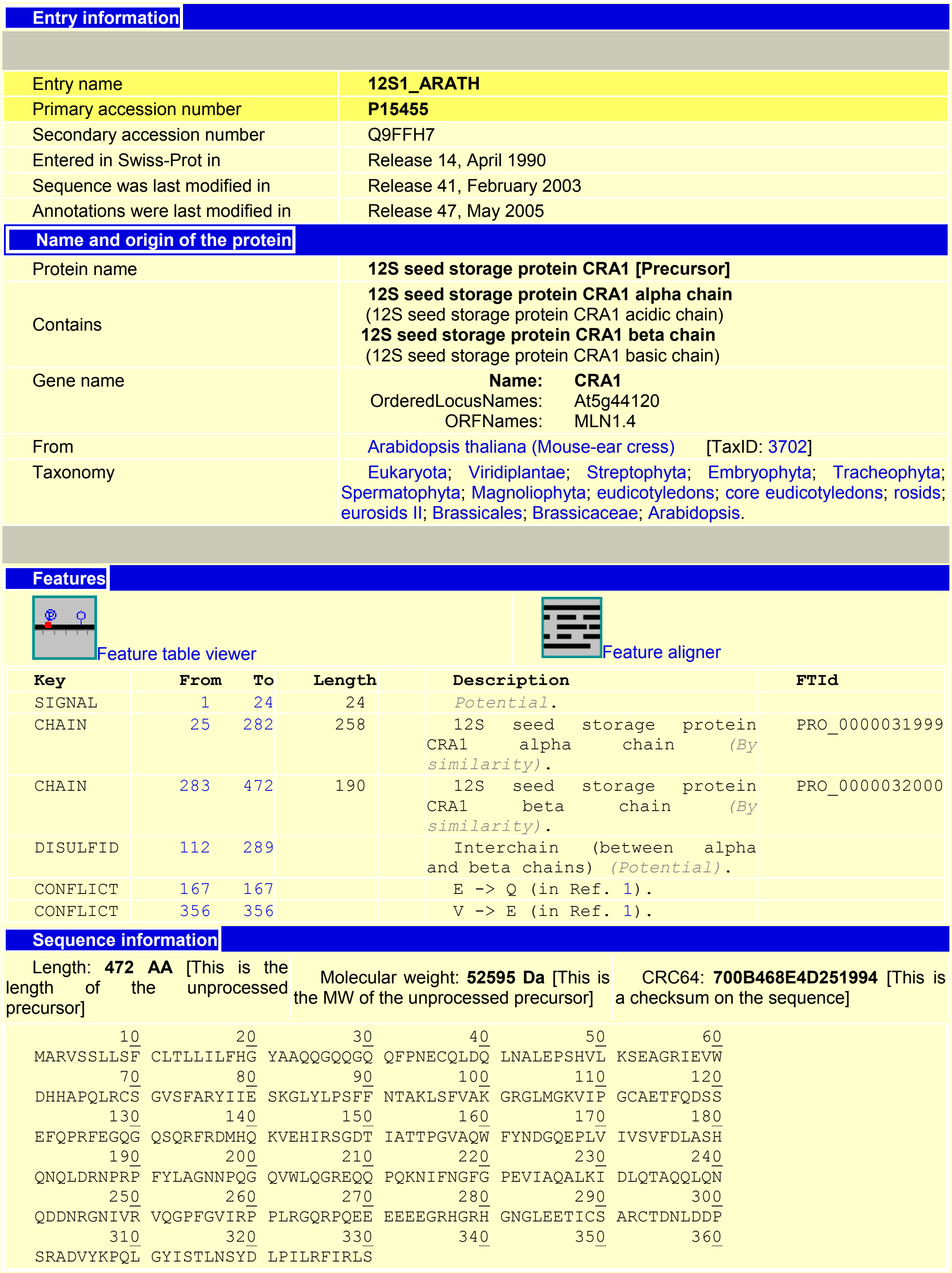
SWISSPROT is an annotated protein sequence database. Two classes of data can be distinguished: the core data and the annotation. For each sequence entry the core data consists of:

- The sequence data;
- The citation information;
- The taxonomic data;

The annotation consists of the description of the following items:

- Functional annotations of the protein;
- Posttranslational modifications;
- Domains and sites;
- Secondary structure;
- Quaternary structure;
- Subcellular localizations;
- Similarities to other proteins;
- Diseases associated with any number of deficiencies in the protein;
- Sequence variants, etc.

SWISSPROT can be accessible via www.expassy.org/sprot. An example entry is shown in Figure 31.



Entry information

Entry name	**12S1_ARATH**
Primary accession number	**P15455**
Secondary accession number	Q9FFH7
Entered in Swiss-Prot in	Release 14, April 1990
Sequence was last modified in	Release 41, February 2003
Annotations were last modified in	Release 47, May 2005

Name and origin of the protein

Protein name	**12S seed storage protein CRA1 [Precursor]**
Contains	**12S seed storage protein CRA1 alpha chain** (12S seed storage protein CRA1 acidic chain) **12S seed storage protein CRA1 beta chain** (12S seed storage protein CRA1 basic chain)
Gene name	**Name: CRA1** OrderedLocusNames: At5g44120 ORFNames: MLN1.4
From	Arabidopsis thaliana (Mouse-ear cress) [TaxID: 3702]
Taxonomy	Eukaryota; Viridiplantae; Streptophyta; Embryophyta; Tracheophyta; Spermatophyta; Magnoliophyta; eudicotyledons; core eudicotyledons; rosids; eurosids II; Brassicales; Brassicaceae; Arabidopsis.

Features

Feature table viewer Feature aligner

Key	From	To	Length	Description	FTId
SIGNAL	1	24	24	*Potential*.	
CHAIN	25	282	258	12S seed storage protein CRA1 alpha chain *(By similarity)*.	PRO_0000031999
CHAIN	283	472	190	12S seed storage protein CRA1 beta chain *(By similarity)*.	PRO_0000032000
DISULFID	112	289		Interchain (between alpha and beta chains) *(Potential)*.	
CONFLICT	167	167		E -> Q (in Ref. 1).	
CONFLICT	356	356		V -> E (in Ref. 1).	

Sequence information

Length: **472 AA** [This is the length of the unprocessed precursor]

Molecular weight: **52595 Da** [This is the MW of the unprocessed precursor]

CRC64: **700B468E4D251994** [This is a checksum on the sequence]

```
        10         20         30         40         50         60
MARVSSLLSF CLTLLILFHG YAAQQGQQGQ QFPNECQLDQ LNALEPSHVL KSEAGRIEVW
        70         80         90        100        110        120
DHHAPQLRCS GVSFARYIIE SKGLYLPSFF NTAKLSFVAK GRGLMGKVIP GCAETFQDSS
       130        140        150        160        170        180
EFQPRFEGQG QSQRFRDMHQ KVEHIRSGDT IATTPGVAQW FYNDGQEPLV IVSVFDLASH
       190        200        210        220        230        240
QNQLDRNPRP FYLAGNNPQG QVWLQGREQQ PQKNIFNGFG PEVIAQALKI DLQTAQQLQN
       250        260        270        280        290        300
QDDNRGNIVR VQGPFGVIRP PLRGQRPQEE EEEGRHGRH GNGLEETICS ARCTDNLDDP
       310        320        330        340        350        360
SRADVYKPQL GYISTLNSYD LPILRFIRLS
```

Figure 31. SWISSPROT entry for 12S1_ARATH

APPENDIX D. DSSP

The DSSP (Dictionary of Secondary Structure of Proteins) program was designed Kabsch and Sander (1983) to standardize secondary structure assignment. The DSSP database is a database of secondary structure assignments for all protein entries in the Protein Data Bank (PDB). The secondary structure codes used in DSSP are H, alpha helix; B, residue in isolated beta-bridge; E, extended strand, participates in beta ladder; G, 3-helix (3/10 helix); I, 5 helix (pi helix); T, hydrogen bonded turn; S, bend.

In addition to secondary structure assignments, a number of residue properties are given in DSSP. These are entropies, conservation weights, crystal contacts, solvent accessibilities (either in percentage or relative between 0 and 9), B-factors, angle deviations, torsions, Phi/Psi scores, planarities, chiralities, rotamers, bumps, H-bonds, and the number of similar backbone conformations found in the database for all residues of the sequence.

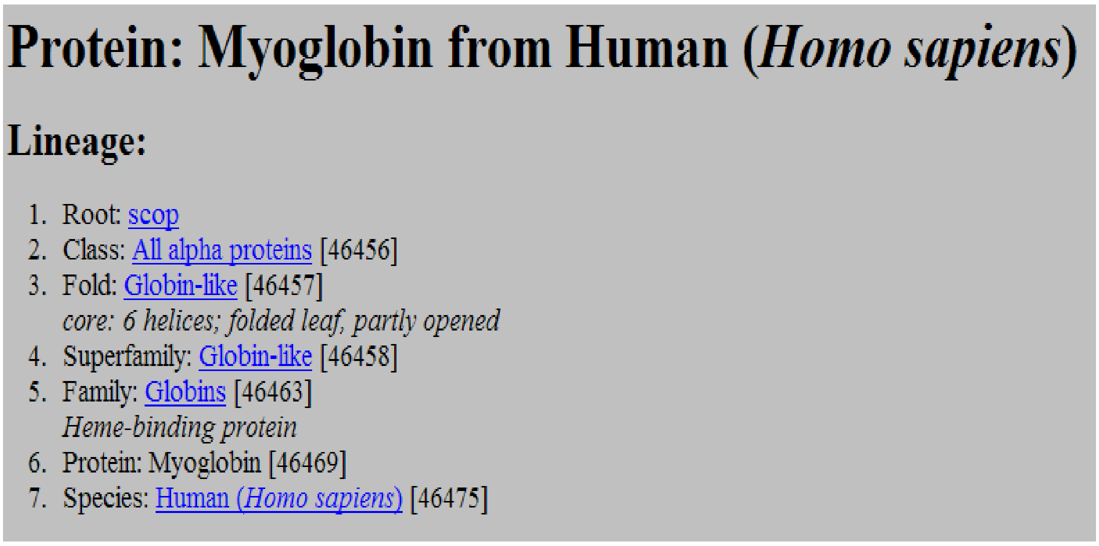
There is no search tool for DSSP, but all files can be downloaded via FTP in text or XML format via http://www.cmbi.kun.nl/gv/dssp. An example entry of DSSP for protein 103L is given in 0.

```
ID           : 103L
Header       : HYDROLASE(O-GLYCOSYL)
 Date        : 1992-09-29
Compound      : Phage t4 lysozyme insertion mutant with ser, leu,
and asp inserted after asn 40, cys 54 replaced by thr, cys 97
replaced by ala, (ins(n40-sld),C54t,C97a)
Source       : Bacteriophage t4
Author       : D.W.Heinz
Author       : B.W.Matthews
Exp-Method   : X
 Resolution  : 1.90
 R-Factor    : 0.182
Ref-Prog     : TNT
HSSP-N-Align : 7
T-Frac-Helix : 0.67
T-Frac-Beta  : 0.08
T-Nres-Prot  : 159
T-Water-Mols : 128
HET-Groups   : 3
 Het-Id      : 900
  Natom      : 4
  Name       : BETA-MERCAPTOETHANOL
 Het-Id      : 173
  Natom      : 1
  Name       : CHLORIDE ION
 Het-Id      : 178
  Natom      : 1
  Name       : CHLORIDE ION
Chain        : _
 Sec-Struc   : 159
  Helix      : 106
  Beta       : 13
   B-Bridge  : 1
   Anti-Hb   : 12
 Amino-Acids : 159
  Break      : 1
 Substrate   : 6
 Water-Mols  : 128
Sequence:MNIFEMLRIDEGLRLKIYKDTEGYYTIGIGHLLTXSLDAAKSELDKAIGRNTNGVITK
DEAEKLFNQDVDAAVRGILRNAKLKPVYDSLDAVRRAALINMVFQMGETGVAGFTNSLRMLQQKRWD
EAAVNLAKSRWYNQTPNRAKRVITTFRTGTWDAYK
DSSP:    HHHHHHHHH   EEEEEE TTS EEEETTEE  - HHHHHHHHHHHHTS  TTB
HHHHHHHHHHHHHHHHHHHHH  TTTHHHHHHS HHHHHHHHHHHHHHHHHHHHT HHHHHHHHTT
HHHHHHHHHSSHHHHHSHHHHHHHHHHHHHHSSSGGG
```

Figure 32. DSSP entry for 103L.

APPENDIX E. SCOP

SCOP (Structural Classification of Proteins) is a database of proteins given as a structural classification in four-level hierarchy (Murzin et al., 1995). The highest level, named as class, provides the information about the secondary structure content of the protein, e.g. all-alpha, all-beta, alpha/beta, alpha+beta,… etc. The second level is fold level. In this level, fold structure of the protein is described. All proteins fit into one of the finite number of folds in this level. The third level, called as superfamily, classifies the proteins with low sequence similarity, which is below 30%, but having structural or functional features suggesting a common evolutionary origin. The lowest level is simply the family in which the proteins having a significant sequence similarity and a clear evolutionary common origin are classified. The organization of SCOP is like a three, on the top of all classes the SCOP root exists. The classification of SCOP is made manually by molecular biology experts. The database is available online in http://scop.berkeley.edu. An example output of SCOP for the protein 2mm1 is given in Figure 33.



Protein: Myoglobin from Human (*Homo sapiens*)

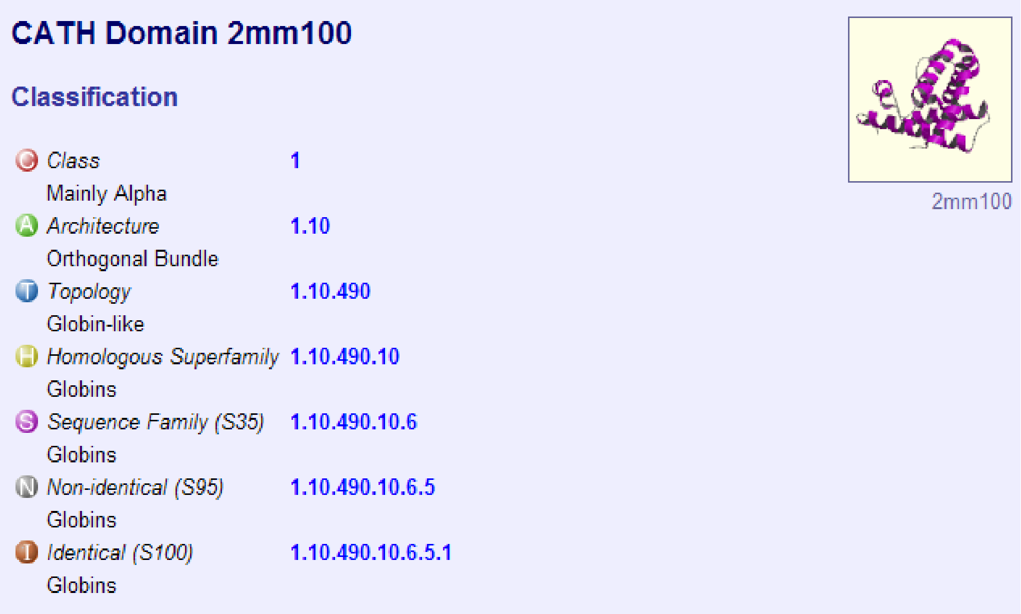
Lineage:

1. Root: scop
2. Class: All alpha proteins [46456]
3. Fold: Globin-like [46457]
   core: 6 helices; folded leaf, partly opened
4. Superfamily: Globin-like [46458]
5. Family: Globins [46463]
   Heme-binding protein
6. Protein: Myoglobin [46469]
7. Species: Human (*Homo sapiens*) [46475]

Figure 33. SCOP output for 2mm1.

APPENDIX F. CATH

CATH (Orengo et al., 1997) is similar to SCOP, but it is partitioned into a different four-level hierarchy. The “class” level is same with the one in SCOP and gives secondary structure composition of the protein. The second level is “architecture” level and the structures are classified based on the overall shape of the domain structures. The third level, called as “topology”, groups the proteins in terms of their fold structures. And the lowest level, named as homology, collects the proteins with high sequence similarity in the same group. The name of the database is coming from the names of the levels; Class, Architecture, Topology, Homology. The construction of the database is semi-automatic, that is, in the highest two levels, some manual constructions are made whereas the low levels are built automatically using similarity considerations. The database is available online in http://www.biochem.ucl.ac.uk. An example output of CATH for 2mm1 is given in Figure 34.



CATH Domain 2mm100

Classification

C	*Class* Mainly Alpha	1
A	*Architecture* Orthogonal Bundle	1.10
T	*Topology* Globin-like	1.10.490
H	*Homologous Superfamily* Globins	1.10.490.10
S	*Sequence Family (S35)* Globins	1.10.490.10.6
N	*Non-identical (S95)* Globins	1.10.490.10.6.5
I	*Identical (S100)* Globins	1.10.490.10.6.5.1

2mm100

Figure 34. CATH output for 2mm1.

APPENDIX G. Pfam

Pfam (Bateman et al., 2002) is a database of protein families. The families are given in the form of alignments and HMM (Hidden Markov Model), which is a probabilistic model to describe the consensus sequences between a group of proteins, profiles. An HMM model consisted of finite number of states which are differentiated as begin state, end state, match state, insert state and delete state. The last three ones are invisible, so called hidden states. Each state has a transition probability associated with it. This probability is position specific in terms of the sequence. The match state has a probability of matching a particular amino acid, which refers to emitting probability. Similarly insert state has a probability associated with each amino acid. The probability of no amino acid associated with a position is represented by the transition probability to a delete state. HMMs are constructed automatically by supervised learning methods based on the multiple alignment results. The database is available online in http://www.sanger.ac.uk/software/pfam.

VITA

Hasan Oğul was born in Bozkır, Konya, 1977. He graduated from METU, Electrical and Electronic Engineering Department in 1998. He received his M.S. degree from Ege University International Computer Institute in 2001. He worked as design engineer in Vestel R&D Unit between 1998-2000. He was a member of Ege University Network Management Group in 2000-2001. Since 2001, he has been working as instructor in Başkent University, Computer Engineering Department, and giving lectures on Data Structures, Programming Languages and Operating Systems. His research interests include Bioinformatics, parallel programming and distributed computing.