ISTANBUL TECHNICAL UNIVERSITY ★ INFORMATICS INSTITUTE

ALIGNMENT AND COMPRESSION-BASED PROTEIN FUNCTION PREDICTION USING SECONDARY STRUCTURE

M.Sc. Thesis by
Aslı FİLİZ

Department: **Advanced Technologies**

Program: **Computer Science**

JUNE 2008

ISTANBUL TECHNICAL UNIVERSITY ★ INFORMATICS INSTITUTE

ALIGNMENT AND COMPRESSION-BASED PROTEIN FUNCTION PREDICTION USING SECONDARY STRUCTURE

M.Sc. Thesis by
Aslı FİLİZ
(704061001)

Date of submission : 5 May 2008
Date of defense examination: 9 June 2008

Supervisor (Chairman): **Assoc. Prof. Dr. Zehra Çataltepe**

Members of the Examining Committee **Assoc. Prof. Dr. Özlem Keskin (Koç University)**

Assoc. Prof. Dr. Hakan Erdoğan (Sabancı University)

JUNE 2008

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

HİZALAMA VE SIKIŞTIRMA TABANLI PROTEİN FONKSİYON ÖNGÖRÜSÜNDE İKİNCİL YAPININ KATKISI

YÜKSEK LİSANS TEZİ

Aslı FİLİZ
(704061001)

Tezin Enstitüye Verildiği Tarih : 5 Mayıs 2008
Tezin Savunulduğu Tarih : 9 Haziran 2008

Tez Danışmanı : Doç. Dr. Zehra Çataltepe

Diğer Jüri Üyeleri Doç. Dr. Özlem Keskin (Koç Üniversitesi)

Doç. Dr. Hakan Erdoğan (Sabancı Üniversitesi)

HAZİRAN 2008

FOREWORD

First of all, I would like to thank to my advisor Assoc. Prof. Zehra Çataltepe for her great support my master's program, both academic and personal. She gave me the opportunity to work in bioinformatics and satisfy my desire for working on genetics by combining it with computer science. She guided my research and provided fundamental help for solving problems that appeared again and again. I also would like to express my thanks to Assoc. Prof. Uluğ Bayazıt for his help on compression related issues and to Assoc. Prof. Özlem Keskin for her help in guiding me and all of our bioinformatics research group through the puzzles of molecular biology.

I also owe thanks to Caner Kömürlü and Eser Aygün with whom I worked in the Bioinformatics Projects coordinated by Assoc. Prof. Zehra Çataltepe. They were always a helping hand by coming over any problems related to the project and my thesis.

I would also acknowledge thanks to Rudi Cilibrasi who provided the CompLearn Toolkit and also several times provided great and fast technical support.

I acknowledge special thanks to TUBITAK for supporting me with the National Scholarship Program for MS Students (2228) during my master's program.

Finally, I would like to thank my parents for their great patience and tolerance I needed while working on my thesis.

Aslı Filiz

June 2008

CONTENTS

ABBREVIATIONS

AA	: Amino acid sequence
SS	: Secondary structure
DSSP	: Definition of secondary structure of proteins
HEL	: H for alpha-helix, E for beta-strand, L for the rest
NCD	: Normalized compression distance
SW	: Smith-Waterman
1NN	: 1-nearest neighbor
k-NN	: k-nearest neighbor
tNN	: Thresholded nearest neighbor
ROC	: Receiver operating characteristic
AUC	: Area under the ROC curve
PDB	: Protein Data Bank
GO	: Gene Ontology
GOA	: Gene Ontology Annotation

TABLE LIST

FIGURE LIST

ABSTRACT

Protein function prediction is one of the most important and difficult problems in bioinformatics. Predicted or actual protein secondary structure, in addition to amino acid sequence, is often used for function prediction.

Usually, alignment scores between amino acid or secondary structure sequences are used to predict protein function. One of the most frequently used alignment algorithms is the Smith-Waterman alignment which is a local alignment algorithm suitable for detecting remote protein similarities. The normalized compression distance (NCD) is another measure of distance that can be used between protein sequences as well as other kinds of data, such as music, text, images, spam filtering, even physics. Smith-Waterman alignment scores and NCD have already been used for function prediction and it has been shown that NCD performs worse than alignment, while combination of NCD and alignment scores outperforms alignment scores only.

In this study, the secondary structure is involved in protein function prediction by using a combined similarity metric that includes both Smith-Waterman alignment and normalized compression distance scores that consider the secondary structure in addition to the amino acid sequence.

HİZALAMA VE SIKIŞTIRMA TABANLI PROTEİN FONKSİYON ÖNGÖRÜSÜNDE İKİNCİL YAPININ KATKISI

ÖZET

Protein fonksiyon öngörüsü, biyoinformatik alanının başlıca zor ve önemli konularından biridir. Amino asit dizisine, yani birincil yapıya, ek olarak tahmin edilmiş veya gerçek ikincil yapı, yani proteinin üç boyutlu yapısının ilk seviyesi, bu problemin çözümünde sıklıkla kullanılmaktadır.

Fonksiyon öngörüsünde genellikle amino asit dizileri ve ikincil yapıların hizalama puanları kullanılmaktadır. Hizalama puanları, protein dizilerinin benzerlik derecesini tespit etmek amacıyla bu dizileri bütünüyle (global hizalama) veya kısmen (yerel hizalama) eşleştirmeye çalışarak eşleşme oranını belirleyen hizalama algoritmaları tarafından, istatistiksel verilere dayanarak hazırlanmış yer değiştirme matrislerine göre belirlenen benzerlik ölçütleridir. En çok tercih edilen hizalama algoritmalarından biri, bir yerel hizalama algoritması olan ve uzak proteinlerin benzerliğinin bulunmasında oldukça başarılı sonuçlar veren Smith-Waterman algoritmasıdır.

Normalize sıkıştırma uzaklığı (NCD) ise proteinlerde olduğu kadar müzik, metin, resim, istenmeyen e-posta filtreleme ve hatta fizik alanından veriler üzerinde de başarılı uygulamaları bulunan diğer bir uzaklık ölçütüdür. NCD, tam olarak hesaplanması mümkün olmayan Kolmogorov uzaklığına bir yaklaşıklık olarak geliştirilmiş ve belirli bir sıkıştırma algoritması kullanılarak sıkıştırılan iki protein dizisinin sıkıştırılmış uzunluklarının, birlikte sıkıştırıldıklarında elde edilen uzunluğa kıyaslanmasına dayanan bir uzaklık, başka bir deyişle benzemezlik ölçütüdür. Kullanıcının belirlemesi gereken bir parametre içermeyen NCD'nin, aynı zamanda kullanılan sıkıştırma algoritmasından da bağımsız, evrensel ve gürbüz bir ölçüt olduğu belirtilmektedir.

Smith-Waterman ve NCD daha önce protein fonksiyon öngörüsünde denenmiş ve Smith-Waterman hizalama puanlarına dayanarak yapılan öngörünün NCD puanları ile yapılan öngörüden daha başarılı olduğu, ancak bu iki ölçütün kombinasyonun, ikisinin tek tek kullanılmasına kıyasla daha iyi sonuç verdiği belirtilmiştir.

Bu çalışmada, her ikisi de amino asit dizisine ek olarak ikincil yapıyı da çeşitli oranlarda dikkate alacak biçimde düzenlenmiş Smith-Waterman hizalaması ve normalize sıkıştırma uzaklığının birleştirilmesi ile elde edilen yeni bir ölçüt kullanılmıştır.

1. INTRODUCTION

Protein function prediction is one of the most important and difficult problems in bioinformatics. Using pattern recognition methods, function prediction deals with the problem of predicting the function of a protein with known structure of different levels, based on a set of proteins whose functions are already known.

Protein structure is defined on four levels, which are the amino acid sequence, secondary structure, tertiary structure and quaternary structure, all of which can be used for bioinformatics applications. The most frequently used ones are the amino acid sequence and the secondary structure since they are less costly to evaluate and hence more available. Besides numerous studies on amino acid sequences, secondary structure has been used for fold recognition by Wallqvist *et al.* (2000), Soeding (2005) and Cheng and Baldi (2006).

Yu and Liu (2004) propose a correlation based feature selection algorithm called the Fast-Correlation Based Filter (FCBF) which is also applicable to bioinformatics data sets where the number of features is usually very large. Çataltepe *et al.* (2007) compare FBCF to two other dimensionality reduction algorithms, principal component analysis (PCA) and Fisher's linear discriminant analysis (Fisher's LDA) using different classification algorithms and show that FCBF either significantly increases or just slightly decreases the classification accuracy whereas other dimensionality reduction techniques lead to dramatic decreases.

A popular approach of using structural similarities in protein function prediction is using alignment-based classification. Alignment is matching similar parts of biological data such as gene sequences or protein structure sequences. The most frequently used alignment algorithms are the Needleman-Wunsch global alignment algorithm (Needleman and Wunsch, 1970) and the Smith-Waterman local alignment algorithm (Smith and Waterman, 1981) which is a variation of the first one. Smith-Waterman alignment algorithm is interested in partial matching of sequences; hence it is more suitable for detecting remote protein similarities. Liao and Noble (2003) built pairs of sequences in the data set and obtained pairwise alignment scores by aligning these.

They showed that using the pairwise alignment scores as features of input to support vector machine classifiers is a straight-forward method that outperforms many previous work (Liao and Noble, 2003), e.g. the SVM-Fisher method (Jaakkola *et al.,* 1999 and Jaakkola *et al,.* 2000), the PSI-BLAST algorithm (Altschul *et al.,* 1997), SAM (Krogh *et al.,* 1994) and FPS (Grundy, 1998), especially when working with large data sets. Another work on the contribution of secondary structure to protein function prediction is done by Aygün *et al. (*2008a), where the Smith-Waterman alignment scores are computed by considering the secondary structure in different levels.

The normalized compression distance (NCD) is another measure of distance which is shown to perform quite well in different domains. Keogh *et al. (*2004) present a successful application in pattern recognition, Cilibrasi *et al. (*2004) and Çataltepe *et al. (*2006) made applications in music domain to predict music genre and composer and Cilibrasi and Vitanyi (2005) provide successful implementations of NCD in many areas. There are also application in physics (Benedetto *et al.,* 2002) and spam-filtering (Bratko and Filipic, 2005). Sculley and Brodley (2006) compare different distance metrics using compression, the Chen-Li metric (CLM), the compression-based dissimilarity measure (CDM), compression-based cosine (CosS) and show that NCD outperforms all. Nevill-Manning and Witten (1999) argued that proteins cannot be compressed which was answered by Hategan and Tabus (2004) stating that proteins can be compressed using appropriate compression algorithms. Later Freschi and Bogliolo (2005) applied the LZ78 algorithm for compressing proteins. Li and Vitanyi (1997) and Li *et al.* (2001) show the success of NCD in bioinformatics, especially on classifying genetic data and Ferragina *et al.* (2007) provide another implementation of NCD on biological data.

The NCD was developed by Cilibrasi and Vitanyi (2005) based on Kolmogorov complexity which is not computable, but only approximated. It is a universal, parameter-free (dis)similarity metric which does not depend on the compressor type used. It computes the distance between two sequences, based on their lengths when they are compressed individually or together.

Kocsor *et al.* (2005) compare the success of alignment-based classifiers and compression-based classifiers and shows that using alignment scores only outperforms

using NCD only. However, Kocsor *et al.* suggest a new similarity metric which is a combination of alignment scores and compression scores and report that this new combined metric has a better performance than both alignment (Smith-Waterman and BLAST) and compression (LZW and PPMZ) only.

This study investigates the contribution of secondary structure to protein function prediction both in alignment-based and compression-based methods and suggests a combined similarity metric similar to Kocsor *et al.* which also includes secondary structure. The study on alignment-based classification uses the pairwise Smith-Waterman alignment algorithm and analyses the contribution of different secondary structures to protein function prediction, the results of which are shown by Filiz *et al.* (2008). The study on normalized compression distance includes the suggestion of an NCD metric that encloses the secondary structure additional to the amino acid sequence. The compression scores are computed using the CompLearn Toolkit's LZMA algorithm (Cilibrasi, 2003). Finally, a metric combined of Smith-Waterman alignment scores and normalized compression distance scores, each of which include amino acid sequence and secondary structure, is developed and tested.

The rest of the thesis is organized as follows: Section 2 describes the mostly used features in protein function prediction and explains the data set used in this study in detail. Section 3 explains the alignment-based similarity and normalized compression distance, as well the new combined metric. Section 4 explains the pattern recognition methods used for classification and classifier evaluation. Section 5 reports the experimental results. Section 6 explains the conclusions.

2. PROTEIN FUNCTION PREDICTION

2.1. Definition of Function

As a general heading, function refers to the biochemical role of the protein. Protein function may refer to many things: The biochemical role of the protein within the cell, the cellular function within the tissue the cell belongs to or the structural role within the cell or organism (Petsko and Ringe, 2004).

It is also known that protein function depends on the three-dimensional structure of the protein (Petsko and Ringe, 2004) and a large number of previous works is based on finding associations between the structure of the protein and its function.

2.2. Features Used

2.2.1 Amino acid sequence

Proteins are macromolecules composed of amino acid chains (Tramontano, 2006). An amino acid is a molecule consisting from an amine and a carboxyl functional group, which makes it an acid, a hydrogen atom and a side chain bonded to the alpha-carbon which is the carbon atom the carboxyl groups is also bonded to (see Figure 2.1) (Petsko and Ringe, 2004). The 20 of amino acids found in nature vary only in their side chains. These are isoleucine, alanine, leucine, asparagine, lysine, aspartate, methionine, cysteine, phenylalanine, glutamate, threnonine, glutamine, tryptophan, glycine, valine, proline, arginine, serine, histidine and tyrosine, shown in Table 2.1. with their three-letter and one-letter abbreviations (IUPAC-IUB, 1984).

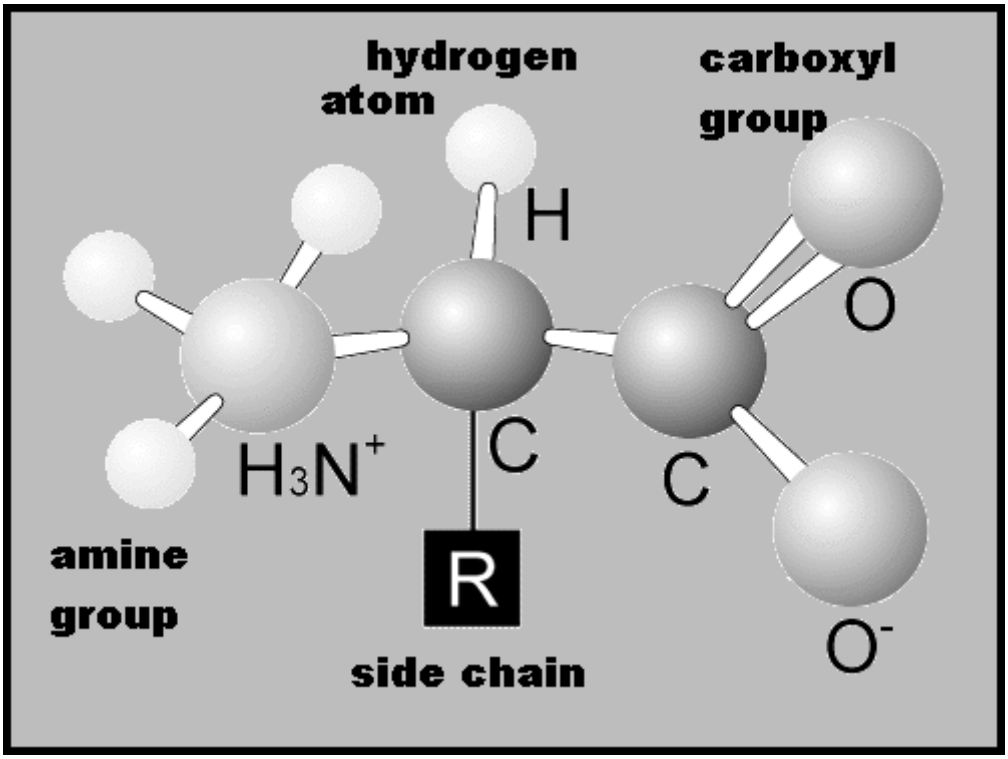

Figure 2.1: Symbolic structure of an amino acid (Plant and Soil Sciences e-Library, 2006)

Table 2.1: Amino acids and their abbreviations (IUPAC-IUB, 1984)

Amino acid	Three-letter abbreviation	One-letter abbreviation
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartic acid (Aspartate)	Asp	D
Cysteine	Cys	C
Glutamine	Gln	Q
Glutamic acid (glutamate)	Glu	E
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanine	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tyrptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V
Any amino acid	Xaa	X

The chain of amino acids or the amino acid sequence (see Figure 2.2) as referred frequently in this study, is the primary structure of a protein (Tramontano, 2006). The amino acid sequence is encoded by the DNA and it is produced with a procedure called *protein biosynthesis* which binds single amino acids with covalent peptide bonds (Petsko and Ringe, 2004).

It is known that the amino acid sequence is closely related to the function of a protein and there are numerous studies on this relationship (Liao and Noble, 2003, Kocsor *et al.* 2005).

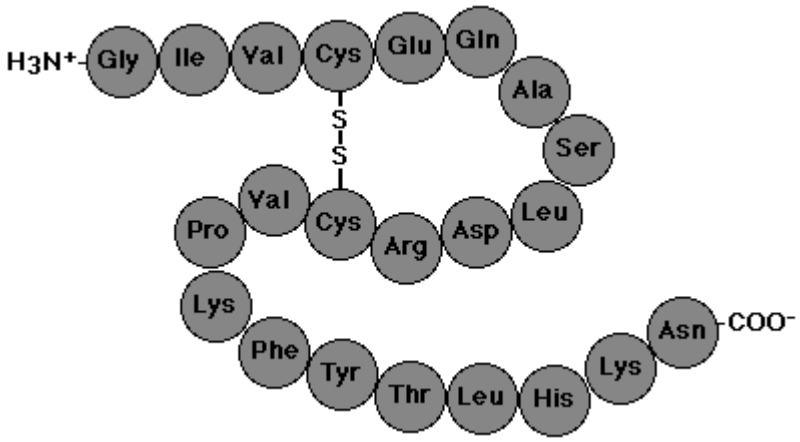

Figure 2.2: Amino acid sequence of a protein, the lysozyme enzyme (Kimball, 2008)

2.2.2 Secondary Structure

The secondary structure of a protein is the three-dimensional form of an amino acid sequence occurring due to hydrogen bonds between amino acids (Petsko and Ringe, 2004). Hence, it is a local structure and different types of secondary structures are seen together in one protein (Petsko and Ringe, 2004).

The alpha-helix is a secondary structure where every amino acid can form hydrogen bonds (Pauling *et al.*, 1951) and the three-dimensional form is a spiral turning to right (Figure 2.3). Alpha helix is mainly formed in regions where the amino acids with alpha - helix preference, Ala, Leu, Met, Phe, Glu, Gln, Lys, Arg, His, are the majority and the other amino acids are not close even if they exist (University of Guelf Department of Chemistry and Biochemistry, 2000).

Figure 2.3: An alpha-helix (University of Miami Department of Biology, n.d.)

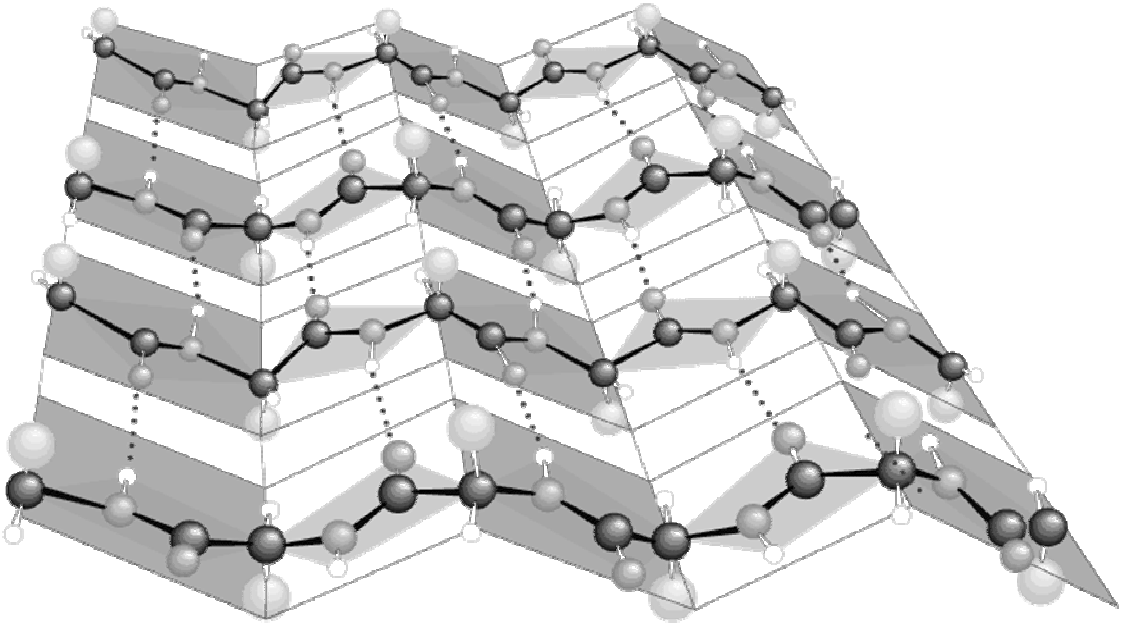

Figure 2.4: Side view of a beta-sheet (Science College, n.d.)

The beta-sheets are secondary structure forms where stretched amino acid strands are placed next to each other so that hydrogen bonds can form between the strands (Pauling and Corey, 1951) which results in a side view of a pleated sheet (see Figure 2.4). These are formed by amino acids with beta-sheet preference, Tyr, Trp, Ile Val, Thr, Cys, and can be either parallel as in Figure 2.5 or anti-parallel as in Figure 2.6 (University of Guelf Department of Chemistry and Biochemistry, 2000).

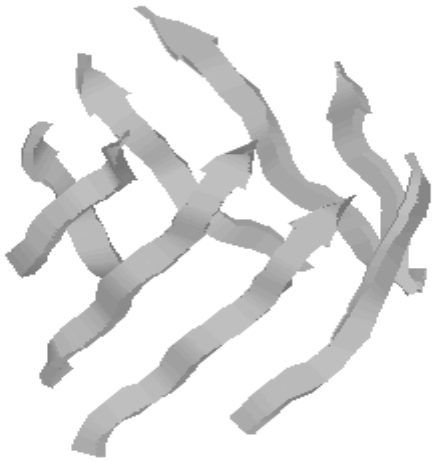

Figure 2.5: Parallel beta sheet (University of Guelf Department of Chemistry and Biochemistry, 2000)

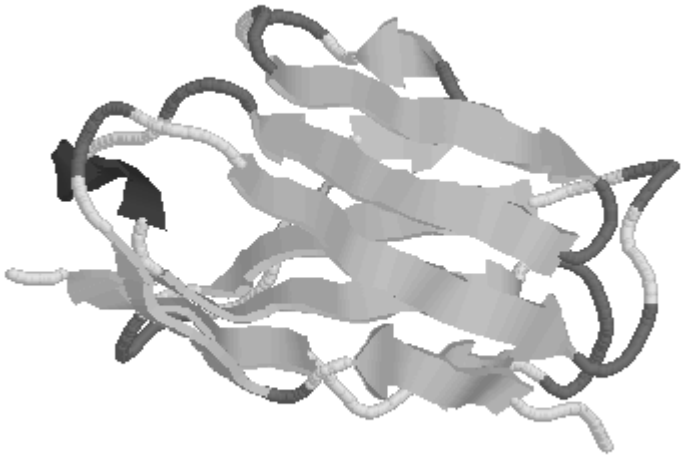

Figure 2.6: Anti-parallel beta sheet (University of Guelf Department of Chemistry and Biochemistry, 2000)

A protein also includes regions that are neither alpha-helices nor beta-sheets. These are called “turns” or “loops”.

The secondary structure is often represented using the DSSP-code introduced by Kabsch and Sander (1983). The 7-letter code is summed up in Table 2.2.

Table 2.2: DSSP representation of secondary structure (Kabsch and Sander, 1983)

DSSP code	Secondary Structure
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 (π-helix)
T	Hydrogen bonded turn
S	Bend

The secondary structure is also used frequently for protein function prediction (Aygün *et al.,* 2008 and Filiz *et al.,* 2008).

2.2.3 Tertiary Structure

Tertiary structure is an irregular structure and is therefore described many ways one of which is the spatial structure of a protein in terms of atomic coordinates (Petsko and Ringe, 2004). It is the composition of secondary structures of one amino acid sequence (Figure 2.7) and is also referred to as “fold” (Petsko and Ringe, 2004).

2.2.4 Quaternary Structure

The last level of protein structure is the quaternary structure which is the compound of more than one amino acid sequence called subunits or monomers (Figure 2.8) (Petsko and Ringe, 2004).

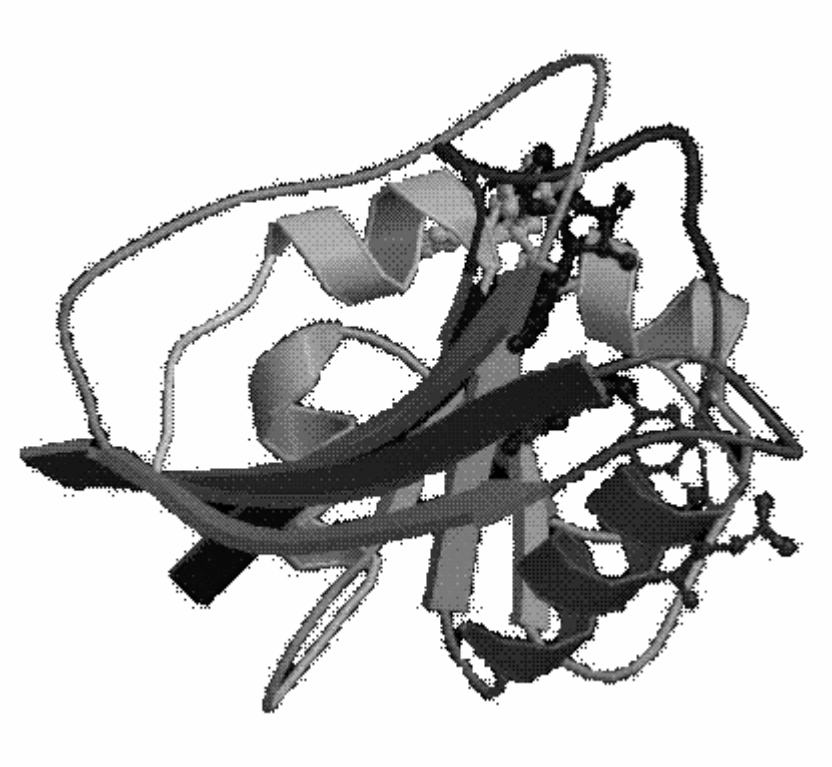

Figure 2.7: Tertiary structure of dihydrofolate reductase (7DFR) (PDB). The spirals are alpha-helices; arrows indicate beta-sheets and the remaining parts that look like threads are loops.

2.2.5 Motifs

In bioinformatics, motif has two meanings.

Firstly, a motif is a partial amino acid sequence that is specific for a certain biochemical function, e.g. the zinc finger motif which is specific for DNA-binding proteins (Petsko and Ringe, 2004).

Secondly, motif is used for a subsequence of the amino acid sequence of a protein which is significant for a function, known as functional motifs, or which acquire a certain secondary structure independent from the neighboring subsequences, known as structural motifs (Petsko and Ringe, 2004).

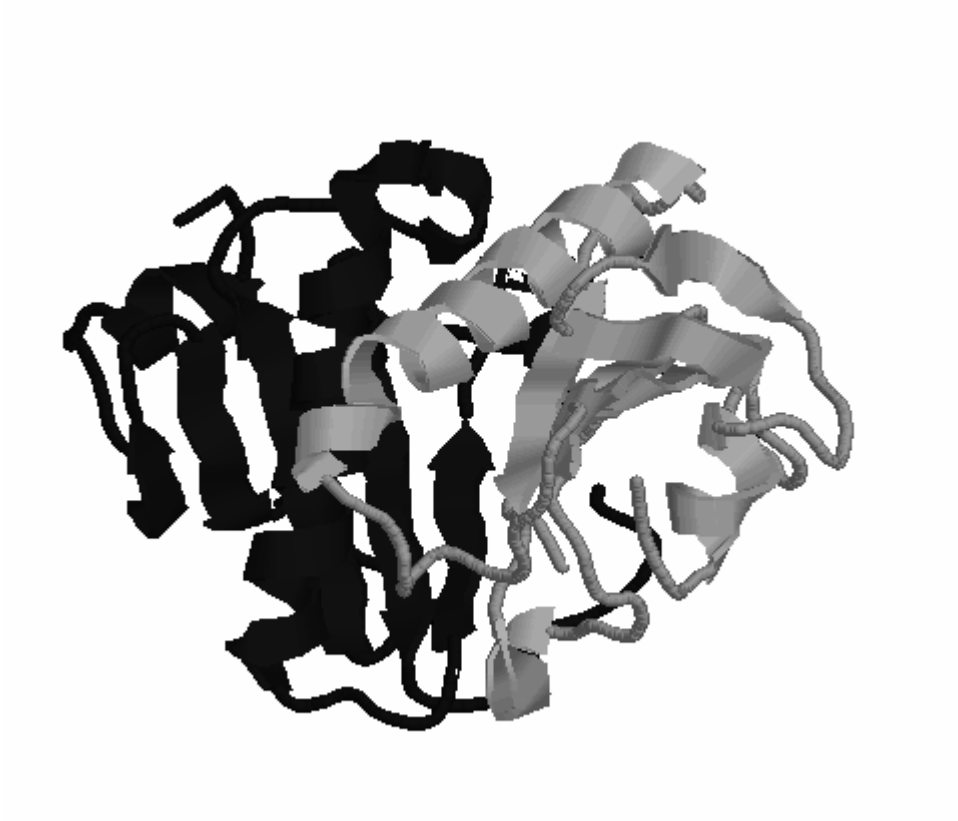

Figure 2.8: Quaternary structure of protein kinase C interacting (1AV5) (PDB)

2.3. Data Set

The data set used in this study consists of amino acid sequences and secondary structure of 4498 annotated proteins, that is to say proteins with known functions. This section explains the development and specifications of the dataset in detail.

2.3.1 Protein Data Bank (PDB), Gene Ontology (GO) and Gene Ontology Annotation (GOA)

The Protein Data Bank (PDB) (Berman *et. al*, 2000) is an online storage for the three-dimensional structures of proteins, nucleic acids and protein-nucleic acid complexes. The PDB founded by Drs. Edgar Meyer and Walter Hamilton at Brookhaven National Laboratory in 1971 containing 7 structures which increased to 50,480 structures in April 2008 (see Figure 2.9). For each structure, sequence details, atomic coordinates, crystallization conditions, 3-D structure neighbors computed using various methods, derived geometric data, structure factors, 3-D images and a variety of links to other resources are available in PDB (Berman *et al.*, 2000).

		Molecule Type				
		Proteins	**Nucleic Acids**	**Protein/NA Complexes**	**Other**	**Total**
Exp. Method	**X-ray**	40066	1025	1823	24	42938
	NMR	6321	805	138	7	7271
	Electron Microscopy	119	11	43	0	173
	Other	88	4	4	2	98
	Total	46594	1845	2008	33	50480

Figure 2.9: PDB Current Holdings Breakdown at 27th April 2008 (NCBI, 2004)

The Gene Ontology (The Gene Ontology Consortium, 2000) is one of the ontologies most frequently used in bioinformatics. It was grounded in 1988 to find consistent annotations for proteins of three organisms: Drosophila melanogaster (fruit fly) from FlyBase, mus musculus (mouse) from Mouse Genome Database and saccharomyces cerevisiae (baker's yeast) from Saccharomyces Genome Database (The Gene Ontology Consortium, 2000). By both many organisms and proteins being added to the database, in January 2008 GO contains over 24,500 terms which include the GO ID, a unique alphanumerical string, the common name and the definition of the protein. GO provides three ontologies: biological processes, cellular components and molecular functions.

The Gene Ontology Annotation (Camon *et al.,* 2004) database provides annotations for proteins of the UniProt Knowledgebase (Butler, 2002) that consists of Swiss-Prot (Boeckmann *et al.,* 2003), TrEMBL (Boeckmann *et al.,* 2003) and PIR-PSD (Wu *et al.,* 2003) using the Gene Ontology (GO). GOA includes GO assignments for the proteins of human, mouse, rat, arabidopsis, zebra fish, chicken and cow.

2.3.2 Retrieval of Annotated Proteins

To obtain a list of annotated proteins, the Gene Ontology Annotation (GOA) project is used (Camon *et al.,* 2004). The Gene Ontology Annotation provides a Protein Data Bank (PDB) (Berman *et al.*, 2000) association file, which contains only the assignments for the proteins present in the PDB database where the structural information is obtained from. To fetch the sequence names, namely the GO IDs, this association file is used.

2.3.3 Homology Reduction

In bioinformatics, homology refers to structural similarity depending on a shared ancestry, even if this original molecule cannot be specified in every case (Petsko and Ringe, 2004). Similarity alone is not enough to determine homology because of the possibility of a similar structure arose by chance (NCBI, 2004).

Homolog amino acid sequences often tend to have similar functions (NCBI2004). This inclination becomes very significant at 40% homology (in other words, when 40% of two sequences are structurally identical) where homolog sequences usually have very similar or identical functions. Prediction of the function of a protein is relatively easier when using its homolog instead of using information obtained from non-homolog proteins, so the high performance of a function prediction algorithm tested on a data set containing homologs can be misleading since it can be caused by homology and not the prediction algorithm itself. Therefore, proteins with 40% sequence identity are removed from the database.

To remove sequence homologs, PDB's scheme is applied. PDB provides several clusterings of proteins generated with CD-HIT or BLASTClust algorithms for different sequence identities. According to the scheme, only the best representative of each cluster is kept for a given clustering. Thereby, potential homologs are removed and non-redundant datasets are obtained. In this study, BLASTClust for 40% identity is used and a dataset of 4498 proteins from human, mouse, rat, arabidopsis, zebra fish, chicken and cow is obtained.

2.3.4 Retrieval of Ontology

The ontology structure is obtained from Gene Ontology (GO) (Ashburner, 1998, Ashburner *et al.,* 2000 and The Gene Ontology Consortium, 2000) database. All the three top level GO classes, molecular function, cellular component and biological process, are included in the data set used in this study. In GO hierarchy, a protein may be associated with more than one term if it is known that it has multiple functions. An example introduced by (The Gene Ontology Consortium, 2000) is cytochrome c which is associated with the molecular function term oxidoreductase activity, the cellular component terms mitochondrial matrix and mitochondrial inner membrane and the biological process terms oxidative phosphorylation and induction of cell death. In such cases, all terms are captured during the labeling process and multi-

labeled data with 1876 dimensional target vectors for all sequences in the data set is generated.

To obtain a well-balanced class distribution, classes with less than 100 or more than 550 sequences are eliminated which resulted in a dataset with 27 classes which are between 2nd and 8th levels in the GO hierarchy (see Figure 2.10 and Figure 2.11). Table 2.3 shows the 27 GO classes used.

Table 2.3: Gene Ontology class distributions in the data set used

Class No	GO ID	Class name	Class type	#Seq
1	0009405	Pathogenesis	Biological process	103
2	0009055	electron carrier activity	Molecular function	105
3	0006810	Transport	Biological process	107
4	0016787	hydrolase activity	Molecular function	117
5	0005506	iron ion binding	Molecular function	118
6	0000166	nucleotide binding	Molecular function	132
7	0003676	nucleic acid binding	Molecular function	137
8	0003700	transcription factor activity	Molecular function	137
9	0006508	Proteolysis	Biological process	148
10	0006412	Translation	Biological process	150
11	0003723	RNA binding	Molecular function	155
12	0008270	zinc ion binding	Molecular function	170
13	0005975	carbohydrate metabolic process	Biological process	173
14	0005179	hormone activity	Molecular function	177
15	0016020	Membrane	Cellular component	202
16	0005515	protein binding	Molecular function	210
17	0005634	Nucleus	Cellular component	214
18	0006355	regulation of transcription, DNA dependent	Biological process	221
19	0005737	Cytoplasm	Cellular component	232
20	0005622	Intracellular	Cellular component	278
21	0005524	ATP binding	Molecular function	288
22	0006118	electron transport	Biological process	297
23	0016491	oxidoreductase activity	Molecular function	300
24	0003677	DNA binding	Molecular function	329
25	0005576	extracellular region	Cellular component	354
26	0008152	metabolic process	Biological process	361
27	0003824	catalytic activity	Molecular function	522
Total				4498

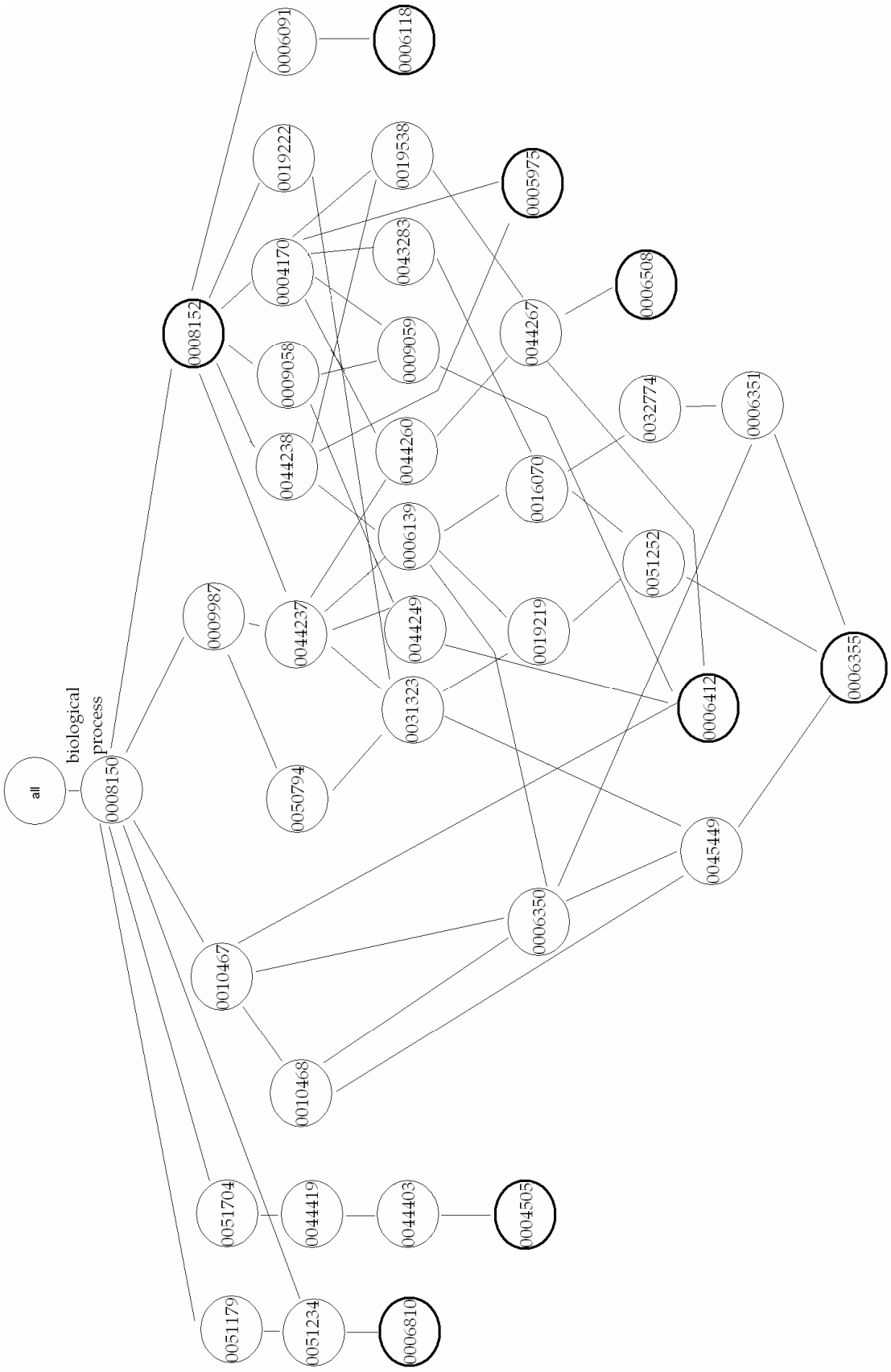

Figure 2.10: GO tree for biological process. Bold circles indicate the classes included in the dataset.

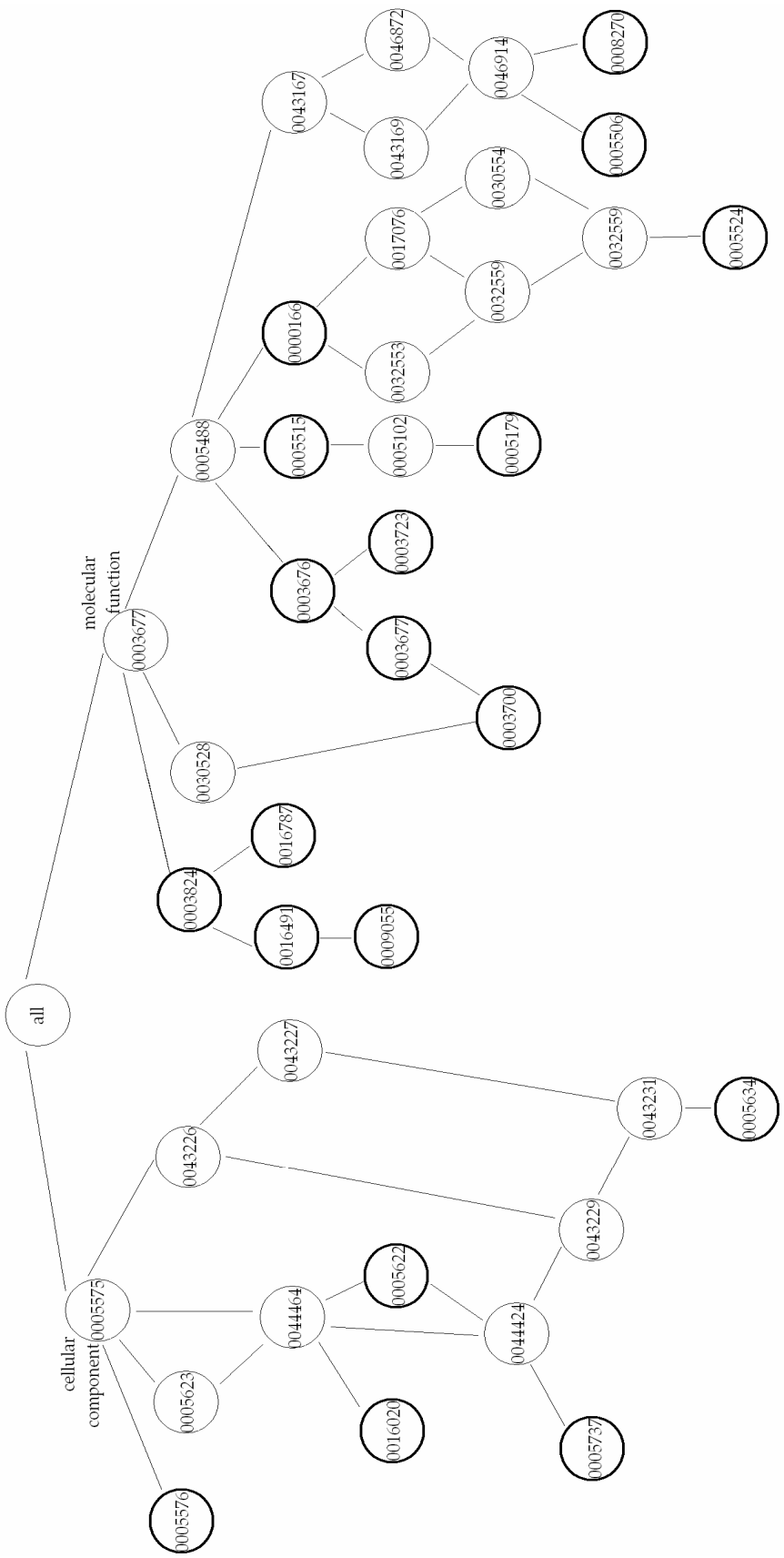

Figure 2.11: GO tree for cellular component and molecular function classes. Bold circles indicate the classes included in the dataset.

2.3.5 Retrieval of Amino Acid Sequence and Secondary Structure

The structure information of each protein downloaded from the PDB web service consists of its amino acid sequence and secondary structure. PDB also provides 3D structure, but it is not included in this work since it does not suggest a protein function prediction scheme specific for PDB but a general scheme including primary and secondary structures.

The PDB provides amino acid sequences as a string of capital letters each of which is the one-letter abbreviation of an amino acid (see Table 2.1). In this study, the amino acid sequences are used just as they are provided by the PDB, in other words without any preprocessing.

The secondary structure provided by the PDB is in DSSP representation. Since this study is concentrated on the contribution of the main three secondary structures, e.g. alpha helices, beta sheets and loops, to the function of the protein, the DSSP representation is converted to the HEL representation (H: alpha helix, E: beta strand, L: loop) according to (Kabsch and Sander, 1983) (Table 2.4).

Table 2.4: Conversion from DSSP to HEL representation (Kabsch and Sander, 1983)

DSSP-code	HEL-code
G, H, I	H
B,E	E
C, S, T	L

The contribution of H, E and L regions to function could rely on their portion (ratio of the length of a specific secondary structure to the length of the whole sequence) in the sequences, since a longer amino acid sequence part provides more structural information than a shorter sequence part. Therefore, for each function class, the average H, E and L portions normalized by the sequence length is calculated. Normalization by the sequence length is necessary since longer amino acid sequences naturally contain longer H, E and L regions and this would lead the relations between H, E and L portions to escape observation. Figure 2.12 and Table 2.5 show the average H, E and L portions in each function class.

Table 2.5: The average ratios of H, E and L regions in GO function classes

Class No	GO ID	Size	H %	E %	L %	Seq. Length
1	0009405	103	21.78 ± 0.2	24.88 ± 0.14	53.34 ± 0.19	184
2	0009055	105	33.44 ± 0.18	16.66 ± 0.12	49.89 ± 0.13	224
3	0006810	107	32.27 ± 0.17	25.75 ± 0.18	41.98 ± 0.14	285
4	0016787	117	32.06 ± 0.11	23.87 ± 0.08	44.07 ± 0.06	299
5	0005506	118	45.06 ± 0.21	12.18 ± 0.13	42.76 ± 0.13	268
6	0000166	132	36.69 ± 0.13	18.59 ± 0.08	44.72 ± 0.11	341
7	0003676	137	29.64 ± 0.13	19.76 ± 0.11	50.60 ± 0.14	278
8	0003700	137	49.32 ± 0.22	9.68 ± 0.11	41.00 ± 0.16	176
9	0006508	148	27.34 ± 0.16	24.11 ± 0.13	48.55 ± 0.14	338
10	0006412	150	29.15 ± 0.19	16.61 ± 0.11	54.25 ± 0.22	242
11	0003723	155	34.62 ± 0.18	19.16 ± 0.12	46.21 ± 0.14	222
12	0008270	170	29.50 ± 0.17	14.73 ± 0.10	55.78 ± 0.18	243
13	0005975	173	30.46 ± 0.15	23.88 ± 0.12	45.77 ± 0.08	420
14	0005179	177	49.25 ± 0.15	4.75 ± 0.06	46.00 ± 0.15	28
15	0016020	202	34.10 ± 0.28	20.92 ± 0.20	44.98 ± 0.17	266
16	0005515	210	32.87 ± 0.24	16.64 ± 0.14	50.49 ± 0.18	236
17	0005634	214	39.18 ± 0.22	12.79 ± 0.13	48.02 ± 0.16	212
18	0006355	221	45.14 ± 0.22	12.53 ± 0.13	42.33 ± 0.16	165
19	0005737	232	38.41 ± 0.13	19.85 ± 0.10	41.74 ± 0.07	368
20	0005622	278	34.59 ± 0.20	14.37 ± 0.11	51.05 ± 0.20	199
21	0005524	288	37.72 ± 0.13	19.50 ± 0.09	42.78 ± 0.09	370
22	0006118	297	37.24 ± 0.18	17.30 ± 0.12	45.46 ± 0.12	313
23	0016491	300	38.30 ± 0.15	19.63 ± 0.10	42.07 ± 0.09	381
24	0003677	329	40.92 ± 0.19	14.30 ± 0.13	44.78 ± 0.14	244
25	0005576	354	37.74 ± 0.23	12.57 ± 0.14	49.69 ± 0.17	99
26	0008152	361	39.96 ± 0.09	18.73 ± 0.07	41.31 ± 0.06	355
27	0003824	522	36.10 ± 0.13	19.67 ± 0.10	44.23 ± 0.10	384
Total		4498				

Table 2.5. also shows the average sequence length in each class. Class 14 includes the shortest sequence with average length 28, which is c.a. 3.5 times shorter than the closest class, which is Class 25 with average sequence length 99. The longest sequences are in Class 13 which has the average sequence length 420; however the average sequence length for the closest class, Class 27, is 383 which is only c.a. 1.09 times shorter than Class 13. Besides, the average ratio of beta-sheet (E regions) in class 14 is c.a. a half of the closest class, Class 8 with 9.68% E regions ratio. Therefore, Class 14 has to be considered as an outlier due to its average sequence length and its E regions ratio, which possibly could affect the classification results.

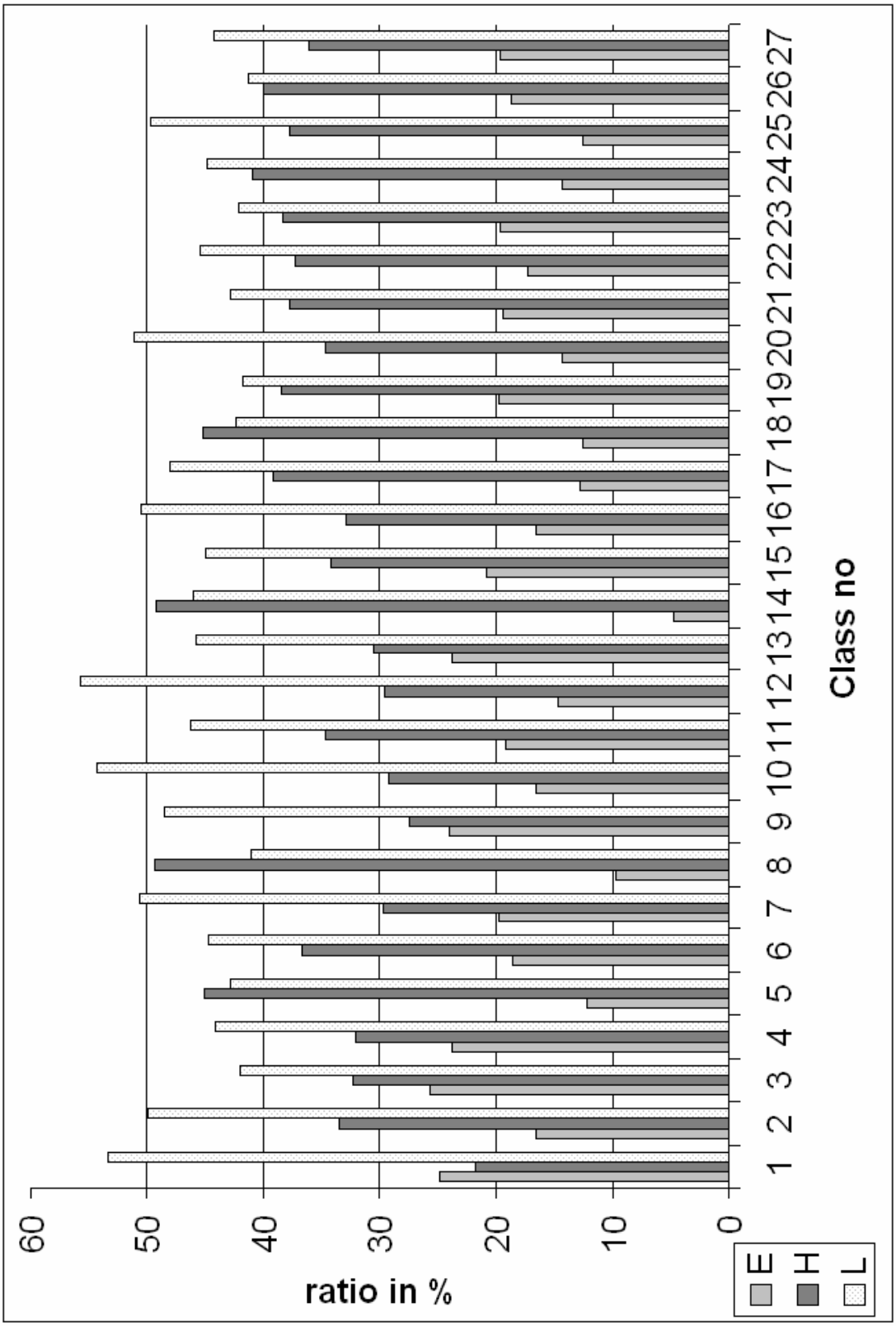

Figure 2.12: The average ratios of H, E and L regions in GO function classes

The statistical significance of the distribution of secondary structures (see Table 2.5) is tested with the analysis of variances. For each class *C*, the following ratio is computed where K indicates the number of sequences in class *C*, *L* indicates the number of statistics ($L = 3$ since the statistics are computed for H, E and L ratios), m_J indicates the mean of the values for the statistic *j* ($j=1$ for H, $j=2$ for E *and* $j=3$ for L), m indicated the mean of all m_J and X_{iJ} is the value of i^{th} sequence for the j^{th} statistic, e.g. X_{12} is the E-ratio in the 1st sequence:

Table 2.6: Significance test using analysis of variance for Table 2.5

Class No	GO ID	ratio(C)	Significance Level		
			0.99	0.95	0.90
1	0009405	94.9	4.605	2.9957	2.30259
2	0009055	139.0	4.605	2.9957	2.30259
3	0006810	22.4	4.605	2.9957	2.30259
4	0016787	160.3	4.605	2.9957	2.30259
5	0005506	155.4	4.605	2.9957	2.30259
6	0000166	208.3	4.605	2.9957	2.30259
7	0003676	211.2	4.605	2.9957	2.30259
8	0003700	208.7	4.605	2.9957	2.30259
9	0006508	130.3	4.605	2.9957	2.30259
10	0006412	174.0	4.605	2.9957	2.30259
11	0003723	127.5	4.605	2.9957	2.30259
12	0008270	315.2	4.605	2.9957	2.30259
13	0005975	154.7	4.605	2.9957	2.30259
14	0005179	703.5	4.605	2.9957	2.30259
15	0016020	59.8	4.605	2.9957	2.30259
16	0005515	166.0	4.605	2.9957	2.30259
17	0005634	245.0	4.605	2.9957	2.30259
18	0006355	240.0	4.605	2.9957	2.30259
19	0005737	316.0	4.605	2.9957	2.30259
20	0005622	305.4	4.605	2.9957	2.30259
21	0005524	409.3	4.605	2.9957	2.30259
22	0006118	309.3	4.605	2.9957	2.30259
23	0016491	329.9	4.605	2.9957	2.30259
24	0003677	378.8	4.605	2.9957	2.30259
25	0005576	379.2	4.605	2.9957	2.30259
26	0008152	1084.0	4.605	2.9957	2.30259
27	0003824	692.9	4.605	2.9957	2.30259

$$ratio(C) = \frac{\left(K * \sum_{j=1}^{L} (m_j - m)^2 \right) / (L-1)}{\left(\sum_{j=1}^{L} \sum_{i=1}^{K} (Xij - m_j) \right) / L * (K-1)} \quad \textbf{(2.1)}$$

For all classes, *ratio(C)* is compared to $F_{\alpha,(L-1),L(K-1)}$ which are obtained from F-distribution tables (StatSoft, 2007) where α is the significance level and if ratio*(C)* > $F_{\alpha,(L-1),L(K-1)}$ then the statistics shown in Table 2.5 are proven to be statistically significant for the significance level *(100 – α)*. The ratios and F values are shown in Table 2.6 and the H, E and L distributions are found to be significant at levels 0.99, 0.95 and 0.90.

3. SEQUENCE-SEQUENCE SIMILARITY/DISTANCE COMPUTATION METHODS

In this study, features used for classification of proteins are the structural information. However, the numbers of amino acids in proteins are very different; in the data set used in this study, it varies between 19 and 1733. Hence, the amino acid sequence or the secondary structure sequence cannot be used as a classification feature; they must be processed to obtain features with a constant number for each protein. The two most preferred and most successful methods for working with biological data are making use of sequence alignment and compression which are explained below in detail.

3.1. Sequence Alignment Similarity

Sequence alignment similarity usually points to homology and functional relationships. Alignment is based on matching the identical or similar sequence parts in proteins, either looking for the similarity of whole protein sequences as in global alignment or looking for partial matching as in local alignment algorithms.

3.1.1 Needleman-Wunsch

Needleman – Wunsch alignment algorithm (Needleman and Wunsch, 1970) is a global alignment algorithm trying to align the whole sequences and is therefore more suitable for data sets containing sequences of nearly equal length. The algorithm maximizes the similarity by finding the "maximum match" for the sequence which is the sequence most amino acids of which can be matched with the other sequence (Needleman and Wunsch, 1970). It is based on dynamic programming and the computation involves a 2-dimensional iterative matrix where every possible alignment of every possible amino acid is represented with an alignment score. The Needleman-Wunsch alignment score is the summation of the scores of matched amino acids reduced by the gap penalties if any gaps are opened during alignment.

3.1.2 Smith-Waterman

Smith-Waterman (Smith and Waterman, 1981) is a local alignment algorithm; it is interested in finding similar sub-regions in longer sequences which do not have to be

similar totally and also which may have varying sequence lengths. Therefore, it is very suitable for detecting the similarity of distantly related proteins, which are also called remote proteins. The Smith-Waterman algorithm is actually a variation of the Needleman-Wunsch algorithm and it uses a substitution matrix which is similar to the matrix involved in Needleman-Wunsch computation, but it is modified by setting the negative alignment scores to zero to enable local alignment.

An example Smith-Waterman alignment of two amino acid sequences is seen in Figure 3.1.

Sequence 1	VSPAGMASGYD
Sequence 2	IPGKASYD
Smith-Waterman Alignment	PAGMASGYD | | || || P-GKAS-YD
Alignment Score	8.6667

Figure 3.1: Smith-Waterman alignment of two amino acid sequences using BLOSUM50 substitution matrix

3.1.2.1 Pairwise Smith-Waterman

Pairwise alignment algorithms do not align a sequence to the whole data set; instead they create pairs of sequences from the data set and compute similarity scores for these pairs. Pairwise alignment scores could be used as input to pattern recognition algorithms to be used for function prediction and whether the two proteins are in the same class or not are the outputs, as in (Cheng and Baldi, 2006).

Another approach is to use the alignment scores to all available training sequences as input. This is the approach taken in (Liao and Noble, 2003) and also in this study. "SVM-pairwise" (Liao and Noble, 2003) takes all sequence pairs in the database and aligns them to each other using the Smith-Waterman local alignment algorithm. This is based on the idea that two proteins belonging to the same class can be aligned similarly to a set of proteins containing both positive and negative instances. Alignment scores are then used as the constant-sized feature vector for a protein. For a training set of N sequences, every protein is aligned to all N sequences, including itself, and it has N features. These features are the input to the classification algorithm. Liao and Noble used this method with SVMs and they indicated that this method is not only easy to use, but also superior to similar algorithms (SVM-Fisher (Jaakkola *et*

al., 1999 and Jaakkola *et al.,* 2000), PSI-BLAST (Altschul *et al.,* 1997), SAM (Krogh *et al.,* 1994) and FPS (Grundy, 1998)) due to its low complexity and outputs with higher accuracy because of learning from negative examples. Liao and Noble (2003) found that SVM-pairwise performs especially well when working with large numbers of protein sequences.

In this study, the balign tool developed by Aygün and Çataltepe (2008) for Bioinformatics Project at ITU is used for computing the pairwise Smith-Waterman scores. Balign produces two types of alignment scores, the percent identity and the bit score, which is the sum of the substitution matrix entries for matches minus gap penalties, normalized with respect to the statistical parameters of the scoring system and is therefore comparable between different alignments (NCBI, 2004).

3.1.2.2 Smith-Waterman incorporating secondary structure

Smith-Waterman algorithm can also align sequences according to their secondary structure and balign (Aygün and Çataltepe, 2008) produces Smith-Waterman scores calculated from secondary structure in HEL format using the BLOSUM50 substitution matrix which is the default substitution matrix of MATLAB Bioinformatics Toolbox. Balign allows including secondary structure according to the parameter α chosen by the user from the interval (0, 1). The Smith-Waterman alignment score including also the secondary structure is then defined as below (Aygün *et al.,* 2008):

$$SW_{\alpha}(x, y) = SW_{AA}(x, y) + \alpha\, SW_{SS}(x, y) \qquad \textbf{(3.1)}$$

where x and y are the sequences to be aligned, $SW_{AA}(x, y)$ is the Smith-Waterman alignment score computed from their amino acid sequences and $SW_{SS}(x, y)$ the Smith-Waterman alignment score computed from their secondary structure.

Another approach is taken to find out the importance of each secondary structure element (H, E, L) for each function, portions of amino acid sequence that has corresponding secondary structure of H or E or L are isolated. Then the amino acid sequences that belong to 6 different secondary structure elements, namely HEL, HE, HL, H, E and L are produced. Figure 3.2 shows the original amino acid sequence, secondary structure and each of the six amino acid sequences produced for HEL, HE, HL, H, E and L regions. When a secondary structure element is not used, in the amino acid sequence, the actual residue is replaced by the "+" symbol. BLOSUM50

substitution matrix is modified to incorporate the "+" symbol and the parameter α is set to 0. The Smith-Waterman alignment scores are computed using these sequences, called SW_{HEL}, SW_{HE}, SW_{HL}, SW_H, SW_E *and* SW_L respectively.

Original sequence	QYKEVNETKWKMMDPILTTSVPVYSLKVDKEYEVRVRSKQRNSGN
Secondary structure	HHHHHEEEEELLEHLLEEEEEELLLLLLLLLLLLLLHHHEEEEL
HEL	QYKEVNETKWKMMDPILTTSVPVYSLKVDKEYEVRVRSKQRNSGN
HE	QYKEVNETKW++MD++LTTSVP++++++++++++++SKQRNSG+
HL	QYKEV+++++KM+DPI++++++VYSLKVDKEYEVRVRSKQ++++N
H	QYKEV++++++++D++++++++++++++++++++++SKQ+++++
E	+++++NETKW++M+++LTTSVP++++++++++++++++++RNSG+
L	++++++++++KM++PI++++++VYSLKVDKEYEVRVR+++++++N

Figure 3.2: Secondary structure filtering

3.1.2.3 Conservation score

Conservation score is the normalized version of bit score which is computed as follows:

$$cons(x,y) = \frac{bitscore(x,y)}{\max\left(bitscore(x,x), bitscore(y,y)\right)} \quad \textbf{(3.2)}$$

where *cons(x,y)* is the conservation score of sequences *x* and *y* and *bitscore(x,y)* is the bitscore of sequences *x* and *y*.

Conservation score gives a better measure of similarity due to normalization and is therefore preferred to raw Smith-Waterman alignment scores in this study.

3.2. Normalized Compression Distance (NCD)

Normalized compression distance (NCD) is a parameter-free, universal metric for sequence similarity developed by Cilibrasi and Vitanyi (2005) which is robust to compressor changes and has applications in several research areas, also in bioinformatics.

3.2.1 Distance and metric

A distance function *D* is a metric if it satisfies the following properties with *D(x,y)* being the distance between *x* and *y* (Cilibrasi and Vitanyi, 2005):

1. $D(x,y) = 0$ iff $x = y$
2. **Symmetry:** $D(x,y) = D(x,y)$
3. **Triangle inequality:** $D(x,y) \leq D(x, z) + D(z, y)$.

A frequently used distance is the Euclidean distance which is a metric in the sense defined above. These properties are also applicable to similarity metrics.

3.2.2 Admissible distance

An admissible distance *D(x, y)* is the length of a binary prefix codeword which computes the sequence *x* from the sequence *y* and also the sequence *y* from the sequence *x* using a certain programming language, also called the reference programming language (Cilibrasi and Vitanyi, 2005). The two-way computation makes the admissible distance symmetric. An admissible distance does not have to be a metric, but there are examples like the Hamming distance which are both an admissible distance and a metric (Cilibrasi and Vitanyi, 2005).

3.2.3 Normalized admissible distance

Normalized admissible distance is a similarity distance developed on the assumption that long sequences different only in a short region are much more similar than short sequences differing in a region of the same length (Cilibrasi and Vitanyi, 2005). Therefore, the admissible distance *D* is normalized by another admissible distance D^+ which is defined as follows (Cilibrasi and Vitanyi, 2005):

$$D^+(x,y) = \max\{\max\{D(x,z): C(z) \leq C(y)\}, \max\{ D(z,y): C(z) \leq C(x)\}\} \quad \textbf{(3.3)}$$

where *C* a certain compressor, also called the reference compressor, *C(.)* is the compressed length obtained using the reference compressor *C* and *z* is any sequence.

The normalized admissible distance is a similarity distance, in other words, it shows how distant the sequences are. Therefore, it is often named a dissimilarity or disparity metric.

3.2.4 Kolmogorov complexity

Kolmogorov complexity, a definition from information theory, provides a basis for most of the alignment-free methods of sequence comparison (Kocsor *et al.,* 2005). The conditional Kolmogorov complexity $K(x \mid y)$ is the length of the shortest binary program which computes the sequence x from the sequence y using a universal Turing machine (Li and Vitanyi, 1997).

The non-conditional Kolmogorov complexity $K(x)$ is the same as $K(x \mid \lambda)$ where λ denotes the empty sequence, that is, $K(x)$ is the length of the shortest binary program which computes the sequence x without input using a universal Turing machine (Li *et al.,* 2001).

Using the Kolmogorov complexity, it is possible to produce similarity measures which express the decrease in complexity or conditional complexity (Kocsor *et al.,* 2005). One of them is defined as below in (Li *et al.,* 2001):

$$d_1(x, y) = 1 - \frac{K(y) - K(y \mid x)}{K(xy)} \qquad \textbf{(3.4)}$$

where xy is the concatenation of the sequences x and y.

3.2.5 Normalized information distance

Bennett *et al. (*1998) introduce a new metric called the information distance $E(x, y)$ which is the length of the shortest binary program which computes the sequence x from the sequence y using a universal Turing machine, and vice versa:

$$\mathrm{E(x, y) = \max\{K(x \mid y), K(y \mid x)\}} \qquad \textbf{(3.5)}$$

It is also proven that the information distance is a metric and it is universal since $E(x, y) \leq D(x, y)$ up to an additive constant that is independent from x and y (Cilibrasi and Vitanyi, 2005).

Cilibrasi and Vitanyi (2005) present the normalized information distance $NID(x, y)$ defined as below:

$$NID(x, y) = \frac{\max\{K(x \mid y), K(y \mid x)\}}{\max\{K(x), K(y)\}} \qquad \textbf{(3.6)}$$

NID is also a universal metric. Its weakness is that it is based on the Kolmogorov complexity which is not computable which makes the normalized information distance not computable (Cilibrasi and Vitanyi, 2005). The normalized information distance is also often referred to as the universal similarity metric (USM) (Krasnogor and Pelta, 2004).

3.2.6 Normal compressor

A compressor *C* is normal if it satisfies the following properties up to an O(log *n*) additive term for a sequence of length *n* (Cilibrasi and Vitanyi, 2005):

1. **Idempotency:** C(xx) = C(x) and C(λ) = 0, *λ* being the empty sequence
2. **Monotonicity:** C(xy) ≥ C(x), *xy* the concatenated sequence of *x* and *y*
3. **Symmetry:** C(xy) = C(yx)
4. **Distributivity:** C(xy) + C(z) ≤ C(xz) + C(yz)

3.2.7 Compression distance

The compression distance is an admissible distance which is the approximation of not-computable Kolmogorov complexity by a normal compressor. Being *C* a real-world reference compressor which approximates the properties of normal compressor, the compression distance $E_c(x, y)$ defined as below (Cilibrasi and Vitanyi, 2005):

$$E_c(x, y) = C(xy) - \min\{C(x), C(y)\} \quad \textbf{(3.7)}$$

where $C(x)$ is the compressed length of the sequence *x*.

3.2.8 Normalized compression distance

The normalized compression distance $NCD(x, y)$ is the normalized version of the compression distance $E_c(x, y)$ involving the normal compressor *C* and is defined as follows (Cilibrasi and Vitanyi, 2005):

$$NCD(x, y) = \frac{C(x, y) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}} \quad \textbf{(3.8)}$$

The NCD is a universal, parameter-free similarity distance, since does not need any background knowledge about the data set and is also robust because it is defined independently from the compressor type (Cilibrasi and Vitanyi, 2005).

3.2.9 Compression methods

3.2.9.1 The LZ77 approach

This approach is described by Lempel and Ziv in 1977 and the compressors developed based on this approach make up the LZ77 or LZ1 family (Sayood, 1996). It is an adaptive-dictionary-based technique where the dictionary corresponds to a subset of the already encoded sequence which is examined by a sliding window that consists of a search buffer, which contains a part of the encoded sequence, and a look-ahead buffer, which contains a part of the sequence to be encoded (Sayood, 1996). The LZ77 approach requires no background information on the data to be compressed and is therefore a simple algorithm that assumes that repeating patterns are placed closely on the sequence (Sayood, 1996).

3.2.9.2 The LZ78 approach

This approach is again described by Lempel and Ziv in 1978 and compressors developed based on this approach make up the LZ78 or LZ2 family. This adaptive-dictionary method does not assume the closeness of repeating patterns and therefore uses an explicit dictionary instead of the search buffer that stores the recently encoded part of the sequence (Sayood, 1996).

3.2.9.3 LZMA

The Lempel-Ziv-Markov chain-Algorithm (LZMA) is a variation of the LZ77 that compresses very fast and its compression ratio is 30% greater than that of gzip, another LZ77 variation, and 15% grater than bzip2, also another LZ77 variation. But measuring the compression time related to compression ratio shows that the best compression the LZMA algorithm can make takes 4-12 times longer time than the bzip2 algorithm (7-zip, nd.).

3.2.9.4 Bzip2

bzip2 is a patent-free algorithm developed by Julian Seward that generally compresses the files to 10-15% twice faster at compression and six times faster at decompression than PPM compressors (Bzip.org), however they are much slower than compressors like GNU zip which cannot compress as efficient as bzip2. The algorithm is run in 9 steps that are as follows (Bzip.org): Run-length encoding (RLE) that replaces repeating symbols in a string by its first four characters, Burrows-Wheeler

transform (BWT) which is reversible block-sort algorithm that is essential for bzip2, Move to front (MTF) that identifies frequently repeating strings, Run-length encoding (RLE) that represents the run-length as a binary number, Huffman coding that replaces the binary numbers with codes the length of which depend on their frequency, multiple Huffman tables if the cost of using them does not exceed the cost of including them, unary base 1 encoding for selecting the Huffman tables, Delta encoding that stores each bit length as a difference from the previous one and the sparse bit array (Bzip.org, nd.).

3.2.9.5 GNU zip

GNU zip, or gzip is developed by Gailly and Adler to replace the algorithm *compress* of Linux which causes patent problems. Like compress, gzip is a variation of the LZ77 algorithm (Lempel and Ziv, 1977) and the Huffman coding (Gzip.org, n.d.).

Zlib is also developed by Gailly and Adler and works on gzip-formatted data. When a string has occurred for the second time, it is replaced by a pointer to the previous occurrence using a hash table including all previously seen strings of 3 bytes length. Previous occurrences can be searched within the recent 32KB starting from the closest occurrence to benefit from the Huffman coding (Gailly, n.d.).

3.2.10 CompLearn

In this study, to compute the normalized compression distance scores the CompLearn Toolkit (Cilibrasi, 2003) developed by Cilibrasi, Cruz and De Rooij is used. It is an open source toolkit built based on Vitanyi and Li's work on compression-based learning algorithms.

To test the validity of the package, a small data set of 50 amino acid sequences is randomly chosen from the data set and the 50x50 matrix of NCD scores is computed using the LZW compression algorithm in MATLAB and using CompLearn's LZMA compressor. The comparison of the two matrices of NCD scores shows that they are consistent with each other, i.e. the scores differ only in a constant additional term which will be eliminated by when computing the distances of NCD vectors by the 1NN classifier (see Section 4.1.1). The main difference in the scores appears along the diagonal of the NCD matrices: *NCD(x,x)* computed by CompLearn is dramatically greater than the one computed by the LZW of MATLAB. However this is not a decisive factor, since the normalized compression distance of a sequence to itself is

never needed for testing the classifiers, testing involves only distance of test instances to train instances. Therefore, the CompLearn Toolkit is preferred due to its lower time-complexity when compared to MATLAB.

To reduce the computational complexity, CompLearn makes an assumption in the denominator of the equation (3.8) and uses *C(x)* instead *max{C(x), C(y)}*. Hence, *NCD(x, y)* ≠ *NCD(y, x)* and the developers of the CompLearn package report, that it was experimentally shown that this assumption does not cause any important change in the classification results.

3.2.11 NCD Incorporating Secondary Structure

Normalized compression distance is naturally also computable for sequences of secondary structure. But since the study aims to involve both the amino acid sequence and the secondary structure, new approaches are necessary.

The first approach is a composite NCD score that considers the amino acid sequence and the secondary structure in varying ratios. The NCD scores for the amino acid sequence ($NCD(x_{AA}, y_{AA})$) and the NCD scores for the secondary structure ($NCD(x_{SS}, y_{SS})$) are computed separately and joined with a user defined parameter β:

$$NCD_{\beta}(x, y) = (1 - \beta).\ NCD(x_{AA}, y_{AA}) + \beta\ .\ NCD(x_{SS}, y_{SS}) \qquad \textbf{(3.9)}$$

3.2.12 NCD Using Joint Representation

The second approach is generating a joint representation where each letter stands for an amino acid with a certain secondary structure. Since there are 20 amino acids and 3 secondary structures, the joint representation requires a mapping to an alphabet of 60 characters. This mapping is shown in Table 3.1 and Figure 3.3. show the conversion to the joint representation for the beginning region of the protein 10MH:A according to Table 3.1. The NCD scores are computed using this joint representation and represented as NCD_{60}.

Amino Acid Sequence	MIEIKDKQLTGLRFIDLFAGLGGFRLALESCGAECVYSNEWD
Secondary Structure	LLLLLLLLLLLEEEEELLLLHHHHHHHLLLEEEEEELLL
Joint Representation	GXLXaIapdyRdrNWHcOCRdRPMqbAbJvFRCKE17uiL5I

Figure 3.3: Conversion to joint representation of the beginning part of the protein 10MH:A

Table 3.1: Mapping to joint representation

Amino Acid	Secondary structure	Joint rep	Amino Acid	Secondary Structure	Joint rep.
A	H	A	M	H	E
A	E	B	M	E	F
A	L	C	M	L	g
C	H	D	N	H	h
C	E	E	N	E	i
C	L	F	N	L	j
D	H	G	P	H	k
D	E	H	P	E	l
D	L	I	P	L	m
E	H	J	Q	H	n
E	E	K	Q	E	o
E	L	L	Q	L	p
F	H	M	R	H	q
F	E	N	R	E	r
F	L	O	R	L	s
G	H	P	S	H	t
G	E	Q	S	E	u
G	L	R	S	L	v
H	H	S	T	H	w
H	E	T	T	E	x
H	L	U	T	L	y
I	H	V	V	H	z
I	E	W	V	E	1
I	L	X	V	L	2
K	H	Y	W	H	3
K	E	Z	W	E	4
K	L	a	W	L	5
L	H	b	Y	H	6
L	E	c	Y	E	7
L	L	d	Y	L	8

3.3. Combining Smith-Waterman and Normalized Compression Distance

Kocsor *et al.* (2005) worked on comparison of alignment-based and compression-based classification and they reported that alignment-based classification outperforms the compression-based classification, but combining the classification made by combining the alignment and compression scores outperforms both. They suggest combining the two scores with the formula below:

$$F(x,y)=\left(1-\frac{SW(x,y)}{SW(x,x)}\right)NCD(x,y) \quad \textbf{(3.10)}$$

where $F(x, y)$ is the combined similarity score for the sequences x and y. The Smith-Waterman score is normalized by $SW(x,x)$ since $NCD(x, y)$ is also normalized by $C(x)$, an assumption explained in Section 3.2.10 and it is subtracted from 1 since the Smith-Waterman alignment score is a similarity score and the NCD is a distance (dissimilarity) score.

This is suggested for amino acid sequence only, so it must be extended to incorporate the secondary structure. In Section 3.1.1.2 it is explained how to include the secondary structure in the Smith-Waterman alignment score $SW_\alpha(x,y)$ and in 3.2.9 it is explained how to include secondary structure in the NCD score $NCD_\beta(x, y)$. These reveal a combined similarity score, $f_{\alpha\beta}$ for SW_α and NCD_β which is defined as follows:

$$f_{\alpha\beta}(x,y)=\left(1-\frac{SW_\alpha(x,y)}{SW_\alpha(x,x)}\right)NCD_\beta(x,y) \quad \textbf{(3.11)}$$

However, it is possible that the Smith-Waterman alignment and the normalized compression perform differently on sequences with different length. In other words, the performance of alignment-based and compression-based classifications may depend on sequence length and classifying sequences of a certain length with Smith-Waterman alignment scores can be more successful than with NCD scores, whereas NCD performs better on sequences of other lengths.

To inspect the consistency of the Smith-Waterman alignment scores and NCD scores, the counting inversion method is implemented (Kleinberg and Tardos, 2006). The dataset is split into 10 bins according to the length of sequences. The 5 bins with longer sequences (6^{th}, 7^{th}, 8^{th}, 9^{th} and 10^{th} bins) include only a few sequences, so these are discarded. The 5^{th} bin consists of 81 sequences, therefore 80 random samples from each of the 5 bins are chosen and their Smith-Waterman and NCD scores are computed.

Similar sequences are expected to have a higher Smith-Waterman similarity score and lower NCD scores than sequences with less similarity. Because of that, if sequence S_i is closer to the sequence S_j than the sequence S_k, then $SW(S_i, S_j)$ is expected to be

greater than $SW(S_i, S_k)$. In this case, if the NCD scores are consistent with the Smith-Waterman scores, which means that if they are indicating the same relations between the sequences, $NCD(S_i, S_j)$ must be smaller than $NCD(S_i, S_k)$ or it is an inversion. So, a high number of inversions show that the two scoring algorithms indicate different relationships in the dataset. The pseudocode for counting inversions for the sequence S_i in a data set consisting of N sequences is given in Figure 3.4.

```
for i:1→N
        order SW-scores for Si in decreasing order
        order NCD-scores for Si according to ordered SW-scores
end
for i:1→N
        for j(i+1) →N
                if NCD(Si, Sj) > NCD(Si, Sj)
                        #inversions++
                end
        end
end
```

Figure 3.4: Pseudocode for counting inversions

There is a faster (O(NlogN)) divide-and-conquer algorithm to count inversions; however since our problem size is not too big, we used this straightforward $O(N^2)$ algorithm (Kleinberg, 2006).

For the given bins, Smith-Waterman alignment score for amino acid sequence only (SW_{AA}) and for secondary structure only (SW_{SS}) and NCD scores for amino acid sequence only (NCD_{AA}) and for secondary structure only (NCD_{SS}) are computed. The reference compressor used for NCD is the LZMA compressor of the CompLearn Toolkit.

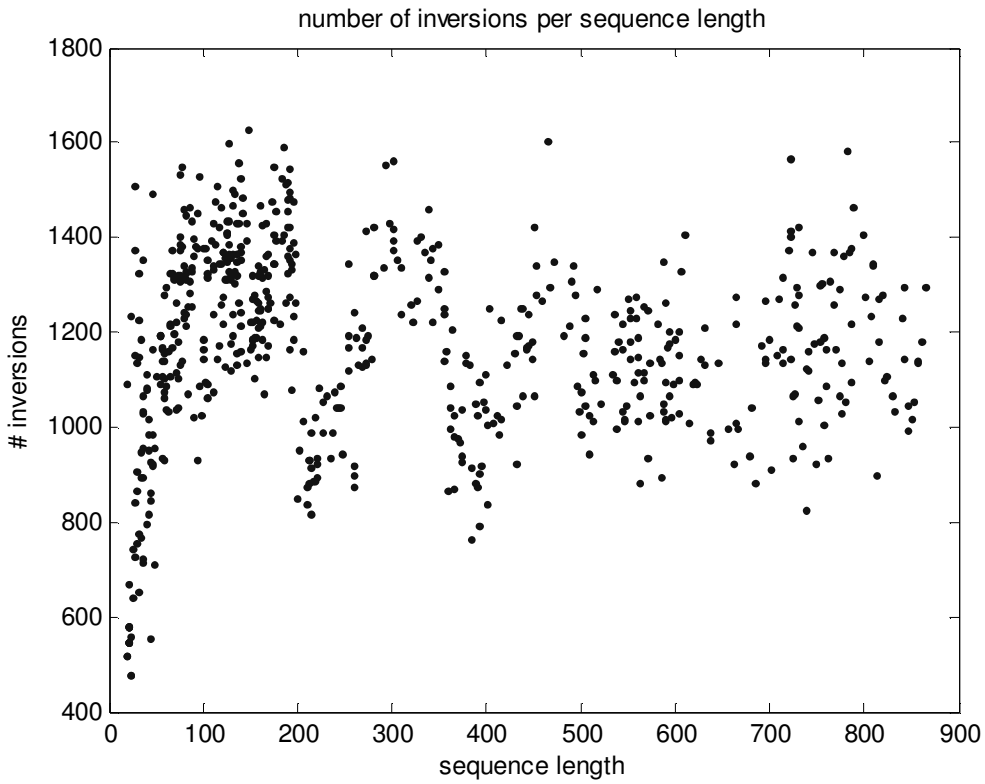

Figure 3.5: Count of inversions for SW_{AA} - NCD_{AA}

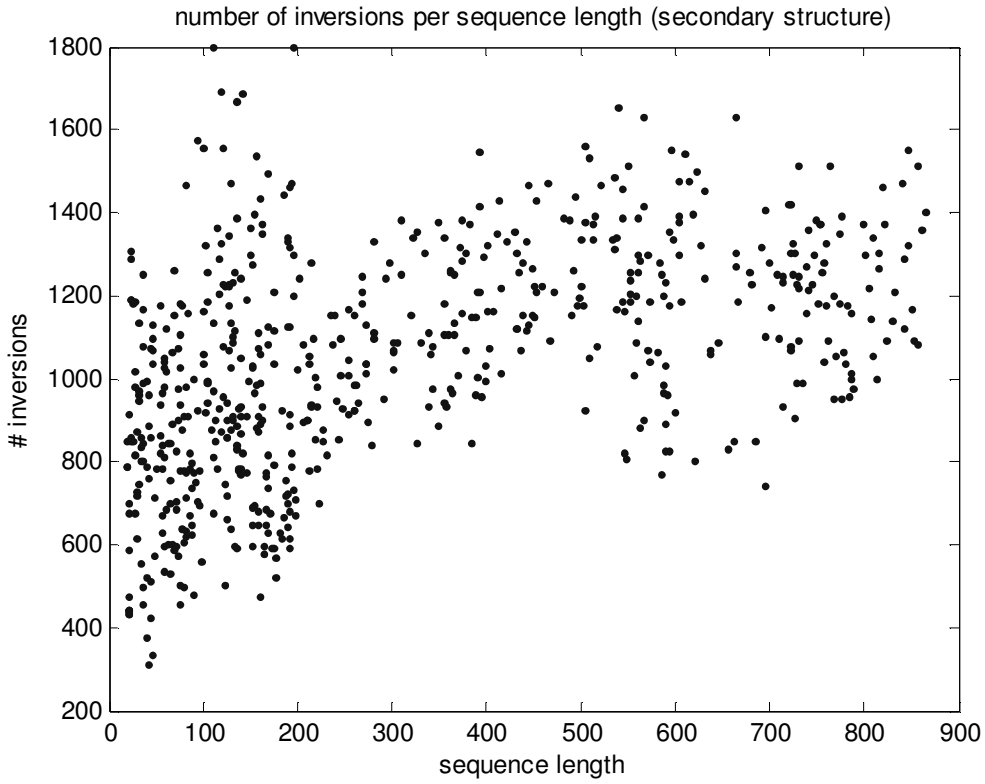

Figure 3.6: Count of inversions for SW_{SS} - NCD_{SS}

Since the counts of inversion computed for amino acid sequences and secondary structures do not reveal any relation (see Figure 3.5 and Figure 3.6), the counts of inversions are normalized by the sequence lengths.

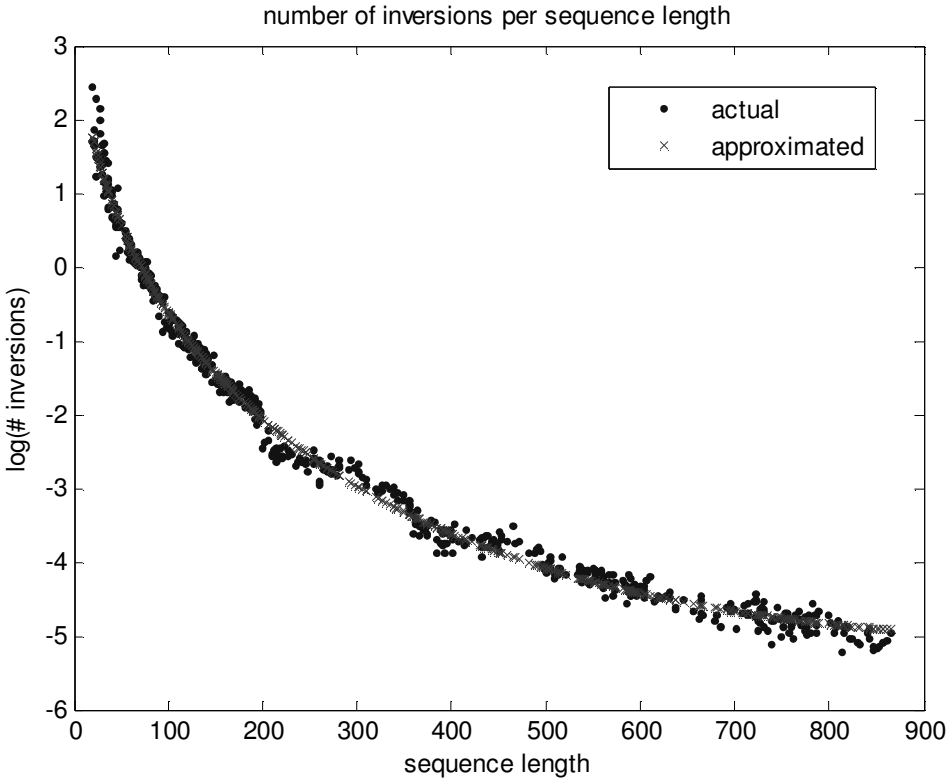

Figure 3.7: Normalized count of inversions for SW_{AA} - NCD_{AA}

The counts of inversions computed for amino acid sequences and for secondary structure show almost the same pattern (see Figure 3.5). The number of inversions decreases exponentially with increasing sequence length and also the standard deviation decreases dramatically with increasing sequence length. For an amino acid sequence and secondary structure of length L, the count of inversion w is related as in Equation (3.11) and Equation (3.12), respectively (see Figure 3.6 and Figure 3.7).

$$\log(w) = 4{,}0365 + 0{,}0083 \,.\, L - 0{,}5486 \,.\, \sqrt{L} \qquad \textbf{(3.11)}$$

$$\log(w) = 3{,}8667 + 0{,}0093 \,.\, L - 0{,}5663 \,.\, \sqrt{L} \qquad \textbf{(3.12)}$$

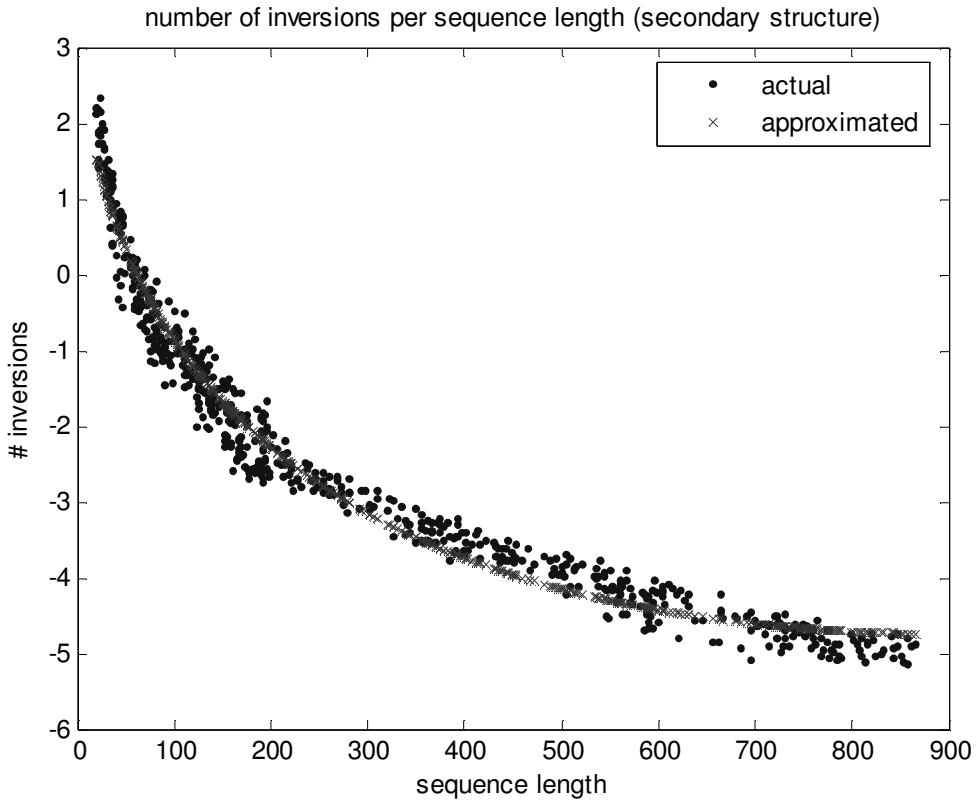

Figure 3.8: Normalized count of inversions for SW_{SS} - NCD_{SS}

The decrease in the number of inversions by increasing sequence length shows that the two metrics indicate the same structural similarities and that these structural similarities are to be recognized much well in longer sequences since the alignment score of two sequences and their compression efficiency is expected to increase with increasing sequence length. For short sequences, the number of inversions is high, which means that one of the metrics fail recognizing the structural similarity of sequences. The failing one is expected to be the compression score since shorter sequences is harder to compress. Therefore, the compression scores should affect the combined similarity score with respect to the sequence length. So, the combined similarity score $f_{\alpha\beta}$ is modified to include normalization with the sequence length:

$$F_{\alpha\beta}(x,y) = \left(1 - \frac{SW_\alpha(x,y)}{SW_\alpha(x,x)}\right)\left(\varphi.NCD_\beta(x,y)\right) \qquad \textbf{(3.13)}$$

where φ is the normalization factor with:

$$\varphi = \left(\frac{\min\{|x|, |y|\}}{\max\{|s|, s \in dataset\}} \right) \quad \textbf{(3.14)}$$

To manually regulate the contribution of the secondary structure, another additive parameter, δ, is introduced into the equation (3.12) and the following formula is obtained:

$$F_{\delta}(x, y) = \left(1 - \frac{SW_{\alpha}(x, y)}{SW_{\alpha}(x, x)}\right)\left((\varphi + \delta).NCD_{\beta}(x, y)\right) \quad \textbf{(3.15)}$$

Section 3.2.12 explains another NCD score, NCD_{60}, for which another combined similarity score has to be defined. Aygün *et al.* report that including the secondary in the Smith-Waterman score at level 25%, i.e. setting α to 0.25, leads to the best classification results (Aygün *et al.,* 2008), therefore Smith-Waterman score computed with $\alpha = 0.25$ is used for the computation of the combined score $F_{60:}$

$$F_{60}(x, y) = \left(1 - \frac{SW_{0.25}(x, y)}{SW_{0.25}(x, x)}\right) NCD_{60}(x, y) \quad \textbf{(3.16)}$$

3.4. Using Smith-Waterman and NCD Scores Together

Another method of combining Smith-Waterman and NCD is using both of them by setting them together to a single feature vector. In this case, the feature vector of a protein consists from its pairwise Smith-Waterman scores including both primary and secondary structures (Equation 3.1), NCD scores including both primary and secondary structures (Equation 3.9), and sequence length. An example for a sequence *X* for a train set consisting of *N* sequences is seen in Figure 3.17. The classifier obtained with this feature vector is called F_{ALL}.

Smith-Waterman	**NCD**	**Seq. length**
[SWα(X, Seq_1) … SWα (X, Seq_N)]	[NCD_{β}(X, Seq_1) … NCD_{β}(X, Seq_N)]	L

Figure 3.9: Feature vector of protein sequence X

4. PATTERN RECOGNITION METHODS

4.1. Classification Algorithms

4.1.1 K-nearest neighbor classifier

The k-nearest neighbor (kNN) classification is a supervised learning algorithm that classifies the test instance to the class to which the majority of the k nearest train instances belong (Alpaydın, 2004). "Nearest" means having the smallest distance computed with a certain distance measure, e.g. Euclidean or cosine. The train instances can be both positive and negative, so the kNN algorithm enables learning from negative examples, too. The parameter k is usually chosen an odd number to avoid the case in which the numbers of train instances belonging to two neighboring classes are equal and the classifier cannot decide between them (Alpaydın, 2004). kNN is a quite straight-forward algorithm with low computational complexity and surprisingly good performance (Kocsor *et al.,* 2005).

1-nearest-neighbor (1NN) is a special case of kNN where k is set to 1 and the test instance is classified to the same as class as its nearest neighbor. This is the classifier used in this study with the Euclidean distance measure.

4.1.2 Thresholded nearest neighbor classifier

The thresholded nearest neighbor classifier (tNN) is a variation of the nearest neighbor classifier. Considering that the negative instances in our data set are not proven negatives in all cases (it is possible that a protein show a specific function, but it is experimentally not shown yet), this algorithm deals only with positive instances. It does not use a distance function, instead it decides based on the preferred similarity score. To decide if the test instance s_i belongs to the class C_j, the classifier finds the most similar train instance, s_j, which belongs to C_j and similarity score of s_i and s_j, called D_{ij} is tested on a threshold taken from the user. If $D_{ij} \geq$ threshold, then s_i is classified to the class C_j.

In this study, the tNN is used with the Smith-Waterman alignment scores SW_{HEL}, SW_{HE}, SW_{HL}, SW_H, SW_E *and* SW_L which are explained in Section 3.1.1.2.

4.1.3 Support vector machines

Support vector machines are discriminant-based supervised learning algorithms learning from the linear discriminant. It is assumed that the classes are linearly separable from each other. The linear discriminant can be used even if no assumptions on class densities in the data set are possible (Alpaydın, 2004).

SVMs define hyperplanes that separate the classes in the data set. The distances between the hyperplanes, or the distance between the instances closest to the hyperplane, are called the margin (Alpaydın, 2004). SVM obtains support vectors by finding the optimal hyperplanes, which means by maximizing the margin.

If the data set is not linearly separable, it can be transformed to a new space of higher dimension where it is linearly separable and this transformation is done by the kernel function (Alpaydın, 2004). The most frequently used kernel functions are polynomial, radial-basis and sigmoidal, whereas homogenous, exponential, cosine, minkowksi etc. kernel functions are also possible.

In this study, SVM classifier is not used, because its performance on a smaller data set with radial-basis kernel was found to be very close to that of 1NN and running SVM on our data set has a high computational complexity.

4.2. One-Against-All

One-against-all classification is a method where the data set is divided into two subsets, the first subset is the class to be predicted and the second subset is made up by all the other classes. The aim is to correctly isolate one certain class from the others.

The dataset used in this study is multi-labeled since an instance may belong to more than one class, which means that each of the 27 classes have to be predicted independently from all other classes. Therefore, for the prediction of each class, it is set to be the class to be predicted and a one-against-all classification is done.

4.3. Classifier Evaluation Methods

4.3.1 K-fold cross validation

On large data sets, the train and test sets can be obtained by randomly partitioning the data set. However, the data sets are usually not large enough for this and each instance is too substantial to be isolated from training or testing.

K-fold cross validation is a method of partitioning the data set which enables to use all instances both for training and testing without damaging the evaluation of the classifier. The data set is be divided into *k* random parts, in other words, *k* subsets of randomly selected sequences are produced (Alpaydın, 2004). On each fold, *(k -1)* subsets make up the train set and the remaining subset is used for testing (Alpaydın, 2004).

In this study, k-fold cross validation with *k=10* is used where each of the 10 subsets have the same class distribution as the original (not partitioned) data set.

4.3.2 Accuracy

To analyze the success of the classification, accuracy is a frequently used measure. It is based on the class confusion matrix which is a table showing how many instances from each class are classified to which class by the classification algorithm used. An example is seen in Table 4.1 where *TP* stands for *true positive*, *TN* for *true negative*, *FP* for *false positive* and *FN* for *false negative*.

Table 4.1: Class confusion matrix (Alpaydın, 2004)

	Predicted Class	
True Class	**Yes**	**No**
Yes	TP	FN
No	FP	TN

Accuracy is defined as follows (Bradley, 1997):

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

4.3.3 Break-even point

Accuracy gives the performance of a classifier at a certain parameter setting. The best accuracy obtained by computing the break-even point, which is the point where recall and precision values are equal to each other (Passerini *et al.,* 2006):

$$precision = \frac{TP}{TP + FP} \quad \textbf{(4.2)}$$

$$recall = \frac{TP}{TP + FN} \quad \textbf{(4.3)}$$

The break-even point analysis is used to determine the best threshold for tNN classifier which is the threshold at the break-even point (Passerini *et al.,* 2006).

4.3.4 Area under the ROC curve (AUC)

Another evaluation technique is the receiver operating characteristics (ROC) curve first used in signal detection theory (Bradley, 1997). The curve which is obtained by drawing the hit rate versus the false alarm rate defined as follows (Alpaydın, 2004):

$$hit\_rate = \frac{TP}{TP + FN} \quad \textbf{(4.4)}$$

$$false\_alarm\_rate = \frac{FP}{FP + TN} \quad \textbf{(4.5)}$$

Each classification algorithm includes a parameter by moving which the number of *TP* and *FP* can also be moved. Increasing *TP*s does also increase the number of *FP*s, so with increasing hit rate, false alarm rate also increases (Alpaydın, 2004) (see Figure 4.1). For random classification, the ROC curve is expected to be the first bisector since hit rate is equal to the false alarm rate, namely both are 0.5. The better the performance of the classifier, the greater is the curvature of the ROC curve since the hit rate grows much faster than the false alarm rate and the curve get closer to the line *hit rate = 1*.

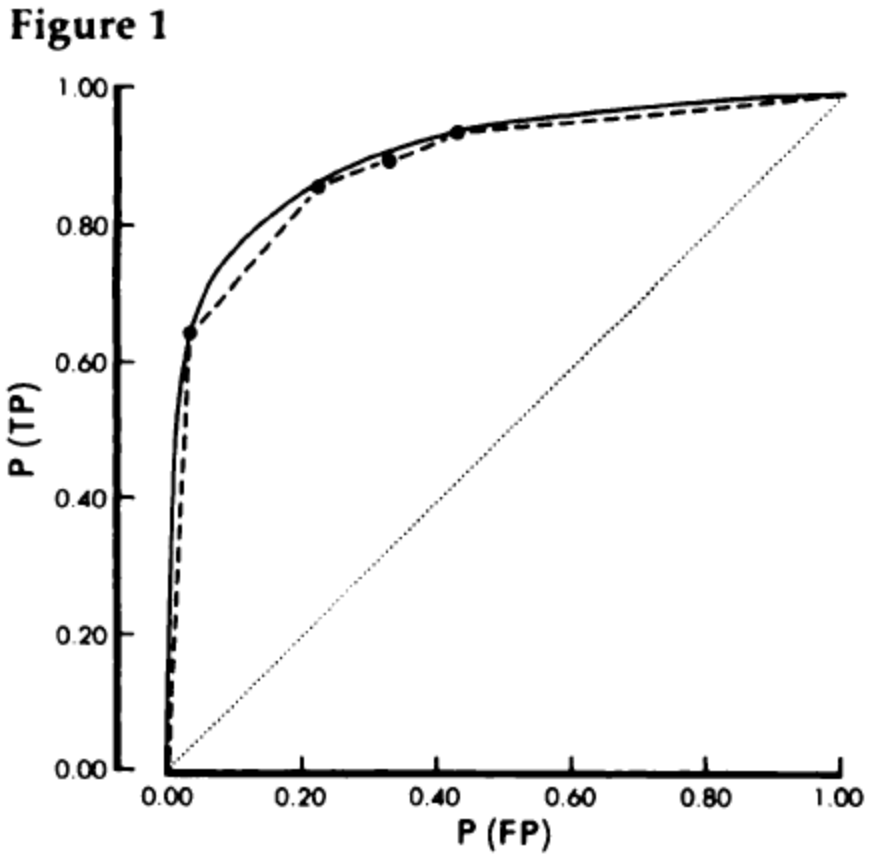

Figure 4.1: An ROC curve (Hanley and McNeil, 1982)

The ROC curve however defines no single operating point; it does not give a single measure independent from decision properties, e.g. thresholds, for comparing to classification algorithms (Bradley, 1997). Therefore, the area under the ROC curve (AUC) is used which is integral of the ROC curve. The better the performance of the classifier, the closer is the ROC curve to the line *hit rate = 1*, so it encloses a greater area. Consequently, the AUC is proportional to the classifier performance.

This study compares the classifier using AUC values obtained from ROC curves computed for each fold of each class. The AUC value for a class is then the mean of the AUC values of its 10 folds. For testing the tNN, the threshold is initially set to the minimum D_{ij} (see Section 4.3) in the test set and moved to the maximum in equally-sized steps. Since AUC is used to determine the classification performance, no certain threshold is needed for tNN.

5. EXPERIMENTAL RESULTS

5.1. Alignment-Based Classification

5.1.1 Classification using amino acid sequence and isolated secondary structure

This section shows the experimental results obtained by using the SW_{HEL}, SW_{HL}, SW_{HE}, SW_H, SW_E and SW_L scores explained in Section 3.1.2.2 in detail. The classification is made using the one-nearest-neighbor (1NN) classifier explained in Section 4.1.1 and the thresholded nearest neighbor (tNN) classifier explained in Section 4.1.2.

5.1.1.1 1NN classification

Function prediction results using the 1NN classification algorithm is shown in Table 5.1. Six classifiers using the SW_{HEL}, SW_{HL}, SW_{HE}, SW_H, SW_E and SW_L scores, called the HEL, HL, HE, H, E and L classifiers respectively, are produced and compared using the AUC values.

For 1NN classification, the HEL classifier, namely the classifier that uses the SW_{HEL} scores or in other words that uses the amino acid regions corresponding to all secondary structure types, has the best performance (mean AUC: 0.90) for all molecular functions except for class 14 (hormone activity) where all classifiers have very close and high performances. The mean AUC of for the HL classifier is 0.86 which is very close to the HEL classifier and it is followed by the H (mean AUC: 0.79) and L classifiers (mean AUC: 0.77). The E (mean AUC: 0.74) and HE classifiers (mean AUC: 0.64) performed generally worse than other classifiers. The higher performance of the classifier using the alpha-helix (H) and loop (L) regions is to be expected since their ratio in the data set is higher than the beta-sheet (E) regions (see Figure 2.12). The fact that the HL classifier performs better than classifiers using H or L regions alone, shows that adding L regions to H regions results in a better prediction. But the facts that the AUC values of the E and HE classifiers are very close to each other and that both are lower than the H classifier indicate that using E regions introduces noise and reduces the classification performance. This also explains the

peak at class 14 since the portions of E regions in this class is only 4,75%, ca. a half of the closest E regions' portion of other classes (see Table 2.5), which makes the sequences in this class far less vulnerable to the noise introduced by E regions.

Table 5.1: Mean AUC values for HEL, HE, HL, H, E and L classifiers using 1NN

		Classifiers					
Class No	**GO ID**	**HEL**	**HL**	**HE**	**H**	**E**	**L**
1	9405	0.86±0.02	0.83±0.02	0.68±0.04	0.62±0.03	0.78±0.02	0.72±0.02
2	9055	0.89±0.02	0.88±0.03	0.73±0.03	0.75±0.03	0.67±0.04	0.78±0.03
3	6810	0.90±0.02	0.86±0.03	0.64±0.03	0.74±0.03	0.78±0.03	0.76±0.04
4	16787	0.94±0.02	0.90±0.02	0.69±0.01	0.84±0.02	0.87±0.01	0.82±0.02
5	5506	0.93±0.01	0.89±0.02	0.66±0.01	0.86±0.02	0.64±0.03	0.75±0.02
6	166	0.92±0.01	0.89±0.02	0.52±0.01	0.83±0.01	0.79±0.02	0.77±0.03
7	3676	0.86±0.01	0.83±0.01	0.61±0.02	0.72±0.02	0.73±0.02	0.72±0.02
8	3700	0.88±0.02	0.85±0.02	0.70±0.02	0.81±0.02	0.52±0.04	0.65±0.02
9	6508	0.93±0.02	0.90±0.01	0.68±0.02	0.81±0.02	0.82±0.02	0.82±0.02
10	6412	0.88±0.01	0.83±0.02	0.58±0.03	0.74±0.02	0.67±0.03	0.75±0.03
11	3723	0.85±0.01	0.82±0.02	0.62±0.02	0.74±0.02	0.67±0.02	0.67±0.04
12	8270	0.91±0.01	0.89±0.01	0.65±0.03	0.73±0.02	0.67±0.02	0.82±0.02
13	5975	0.94±0.01	0.92±0.01	0.69±0.03	0.81±0.02	0.89±0.01	0.82±0.01
14	5179	1.00±0.00	0.99±0.00	0.97±0.01	0.97±0.01	0.95±0.02	0.99±0.00
15	16020	0.85±0.02	0.81±0.02	0.66±0.01	0.66±0.02	0.69±0.02	0.72±0.01
16	5515	0.87±0.01	0.86±0.01	0.55±0.02	0.71±0.01	0.67±0.02	0.73±0.02
17	5634	0.84±0.01	0.81±0.01	0.58±0.02	0.75±0.02	0.61±0.02	0.64±0.02
18	6355	0.88±0.01	0.85±0.01	0.66±0.02	0.77±0.01	0.59±0.03	0.67±0.02
19	5737	0.85±0.01	0.84±0.01	0.48±0.02	0.83±0.01	0.80±0.01	0.77±0.01
20	5622	0.87±0.01	0.84±0.01	0.59±0.01	0.76±0.01	0.64±0.01	0.70±0.02
21	5524	0.91±0.01	0.89±0.01	0.52±0.01	0.85±0.01	0.83±0.01	0.81±0.01
22	6118	0.93±0.01	0.90±0.01	0.66±0.01	0.80±0.01	0.71±0.01	0.83±0.01
23	16491	0.96±0.00	0.94±0.00	0.75±0.01	0.90±0.01	0.84±0.01	0.86±0.01
24	3677	0.88±0.01	0.83±0.01	0.56±0.02	0.76±0.01	0.63±0.02	0.72±0.01
25	5576	0.93±0.01	0.93±0.01	0.82±0.01	0.79±0.02	0.84±0.02	0.88±0.02
26	8152	0.95±0.01	0.93±0.01	0.58±0.02	0.92±0.01	0.89±0.01	0.85±0.01
27	3824	0.94±0.00	0.92±0.01	0.45±0.01	0.88±0.01	0.87±0.01	0.85±0.00
mean		0.90	0.86	0.64	0.79	0.74	0.77

5.1.1.2 tNN classification

Function prediction results using the tNN classification algorithm is shown in Table 5.2. Six classifiers using the SW_{HEL}, SW_{HL}, SW_{HE}, SW_{H}, SW_{E} and SW_{L} scores, called the HEL, HL, HE, H, E and L classifiers respectively, are produced and compared using the AUC values.

tNN has the best performance for the HEL classifier with mean AUC 0.81, followed by the HL classifier (mean AUC: 0.77) as by the 1NN algorithm. Class 14 is again an outstanding point with best performance for each classifier. Different from 1NN, the AUC values for the H, E and L classifiers are very close to each other, mean AUCs 0.66, 0.67 and 0.67 respectively; but the HE classifier performed better than these three classifiers with mean AUC 0.73. Another distinguishing point is the very low AUC values for class 27 (catalytic activity) for all classifiers which is not the case by 1NN except for the HE classifier. The noise effect of E regions stated by the 1NN algorithm is not seen by classification using the tNN algorithm. The performance of the HL classifier being better than the HE classifier is explained by the L portion in the data set which is greater than the E portion. Generally, the AUC values obtained using the tNN algorithm is lower than the AUC values obtained using the 1NN algorithm. Since tNN does not use the negative examples, its lower prediction performance is not surprising as learning from negative examples enhances the prediction performance (Liao and Noble, 2003).

5.1.2 Classification using amino acid sequence and secondary structure on different levels

This section shows the experimental results obtained by using the SW_α scores which is defined with Equation (3.1) in Section 3.1.2.2. The classification is made using the one-nearest-neighbor (1NN) classifier explained in Section 4.1.1. To include the secondary structure in different levels, α is set to 0, 0.25, 0.5, 0.75 and 1.0 and the classifiers obtained are called SW0, SW25, SW50, SW75 and SW100 classifiers respectively. The classification results are shown in Table 5.3 and Figure 5.1.

Aygün *et al.* (2008) experimented using the same methodology on a different data set and found that all classifiers perform very close to each other whereas the SW25 classifier outperforms the other classifiers just slightly. Experimental results for classifiers SW0, SW25, SW50, SW75 and SW100 are consistent with the results obtained by Aygün *et al.* (2008) since the best mean AUC value (0.92) is obtained by the SW25 classifier. This is followed by the SW50 and SW0 classifiers with the mean AUCs 0.91 and 0.90 respectively. The mean AUC value for SW75 is 0.88 and the mean AUC value for SW100 is 0.84 which is the worst performance. For all classifiers, class 14 is an outlier with an AUC of nearly 1.00.

Table 5.2: Mean AUC values for HEL, HE, HL, H, E and L classifiers using tNN

		Classifiers					
Class No	**GO ID**	**HEL**	**HE**	**HL**	**H**	**E**	**L**
1	9405	0.82±0.02	0.76±0.03	0.79±0.02	0.74±0.02	0.65±0.02	0.72±0.03
2	9055	0.84±0.01	0.78±0.02	0.81±0.01	0.68±0.02	0.72±0.01	0.72±0.01
3	6810	0.78±0.04	0.68±0.04	0.73±0.04	0.66±0.03	0.61±0.03	0.62±0.05
4	16787	0.84±0.02	0.74±0.01	0.78±0.02	0.61±0.01	0.70±0.03	0.72±0.02
5	5506	0.85±0.02	0.81±0.02	0.82±0.03	0.72±0.02	0.74±0.02	0.70±0.03
6	166	0.79±0.03	0.70±0.03	0.74±0.03	0.61±0.03	0.64±0.03	0.58±0.02
7	3676	0.77±0.03	0.68±0.03	0.73±0.03	0.65±0.03	0.60±0.04	0.62±0.03
8	3700	0.88±0.02	0.81±0.02	0.86±0.02	0.75±0.02	0.72±0.02	0.75±0.02
9	6508	0.74±0.02	0.61±0.02	0.68±0.03	0.58±0.02	0.54±0.02	0.54±0.03
10	6412	0.84±0.02	0.72±0.01	0.78±0.02	0.65±0.01	0.67±0.02	0.65±0.02
11	3723	0.82±0.02	0.76±0.02	0.78±0.02	0.69±0.02	0.68±0.02	0.68±0.02
12	8270	0.81±0.03	0.74±0.03	0.79±0.03	0.67±0.02	0.69±0.02	0.72±0.03
13	5975	0.75±0.02	0.60±0.02	0.66±0.03	0.53±0.02	0.56±0.01	0.52±0.02
14	5179	1.00±0.00	0.99±0.00	0.99±0.00	0.99±0.01	0.97±0.00	0.99±0.01
15	16020	0.74±0.03	0.69±0.03	0.70±0.03	0.70±0.02	0.62±0.02	0.60±0.03
16	5515	0.80±0.02	0.72±0.02	0.78±0.02	0.68±0.02	0.67±0.02	0.67±0.02
17	5634	0.79±0.02	0.76±0.02	0.77±0.02	0.71±0.02	0.69±0.02	0.69±0.02
18	6355	0.87±0.01	0.81±0.01	0.84±0.01	0.74±0.02	0.72±0.02	0.75±0.01
19	5737	0.67±0.02	0.56±0.03	0.63±0.02	0.48±0.02	0.54±0.02	0.54±0.02
20	5622	0.83±0.01	0.77±0.01	0.79±0.01	0.68±0.01	0.71±0.02	0.66±0.01
21	5524	0.77±0.02	0.64±0.02	0.72±0.02	0.54±0.01	0.56±0.02	0.57±0.02
22	6118	0.81±0.01	0.71±0.01	0.75±0.01	0.60±0.01	0.63±0.01	0.63±0.01
23	16491	0.88±0.02	0.78±0.02	0.82±0.02	0.61±0.02	0.66±0.02	0.68±0.02
24	3677	0.77±0.02	0.69±0.02	0.73±0.02	0.62±0.02	0.64±0.02	0.64±0.02
25	5576	0.93±0.01	0.90±0.01	0.92±0.01	0.88±0.01	0.85±0.01	0.88±0.01
26	8152	0.84±0.01	0.73±0.01	0.79±0.02	0.61±0.01	0.72±0.02	0.69±0.02
27	3824	0.74±0.01	0.60±0.01	0.68±0.01	0.48±0.01	0.51±0.01	0.55±0.01
mean		0.81	0.73	0.77	0.66	0.67	0.67

Table 5.3: Mean AUC values for SW0, SW25, SW50, SW75 and SW100 classifiers using 1NN

		Classifiers				
Class No	**GO ID**	**SW0**	**SW25**	**SW50**	**SW75**	**SW100**
1	9405	0.86±0.02	0.89±0.01	0.88±0.02	0.82±0.02	0.76±0.03
2	9055	0.89±0.02	0.90±0.02	0.91±0.02	0.88±0.02	0.86±0.03
3	6810	0.90±0.02	0.90±0.02	0.90±0.02	0.85±0.03	0.84±0.03
4	16787	0.94±0.02	0.95±0.01	0.96±0.01	0.96±0.01	0.94±0.01
5	5506	0.93±0.01	0.94±0.01	0.94±0.01	0.91±0.01	0.88±0.01
6	166	0.92±0.01	0.94±0.01	0.94±0.01	0.92±0.01	0.90±0.01
7	3676	0.86±0.01	0.91±0.01	0.90±0.01	0.84±0.01	0.78±0.01
8	3700	0.88±0.02	0.89±0.01	0.88±0.01	0.83±0.02	0.77±0.02
9	6508	0.93±0.02	0.94±0.01	0.93±0.01	0.90±0.01	0.86±0.01
10	6412	0.88±0.01	0.90±0.02	0.88±0.02	0.81±0.02	0.72±0.03
11	3723	0.85±0.01	0.87±0.01	0.88±0.01	0.82±0.01	0.80±0.01
12	8270	0.91±0.01	0.93±0.01	0.91±0.01	0.85±0.02	0.79±0.02
13	5975	0.94±0.01	0.95±0.01	0.95±0.01	0.94±0.01	0.92±0.01
14	5179	1.00±0.00	1.00±0.00	0.99±0.00	0.99±0.01	0.97±0.01
15	16020	0.85±0.02	0.87±0.01	0.86±0.02	0.82±0.02	0.77±0.02
16	5515	0.87±0.01	0.89±0.01	0.88±0.02	0.82±0.01	0.77±0.02
17	5634	0.84±0.01	0.86±0.01	0.86±0.01	0.81±0.01	0.77±0.02
18	6355	0.88±0.01	0.89±0.01	0.90±0.01	0.85±0.01	0.80±0.01
19	5737	0.85±0.01	0.88±0.01	0.88±0.01	0.87±0.01	0.86±0.01
20	5622	0.87±0.01	0.89±0.01	0.88±0.01	0.82±0.01	0.76±0.01
21	5524	0.91±0.01	0.93±0.01	0.92±0.01	0.90±0.01	0.88±0.01
22	6118	0.93±0.01	0.94±0.01	0.94±0.01	0.90±0.01	0.86±0.01
23	16491	0.96±0.00	0.97±0.01	0.98±0.01	0.96±0.01	0.94±0.01
24	3677	0.88±0.01	0.90±0.01	0.89±0.01	0.83±0.01	0.78±0.01
25	5576	0.93±0.01	0.95±0.01	0.93±0.01	0.90±0.01	0.87±0.02
26	8152	0.95±0.01	0.96±0.01	0.97±0.01	0.97±0.00	0.96±0.00
27	3824	0.94±0.00	0.95±0.01	0.96±0.00	0.94±0.01	0.92±0.01
mean		0.90	0.92	0.91	0.88	0.84

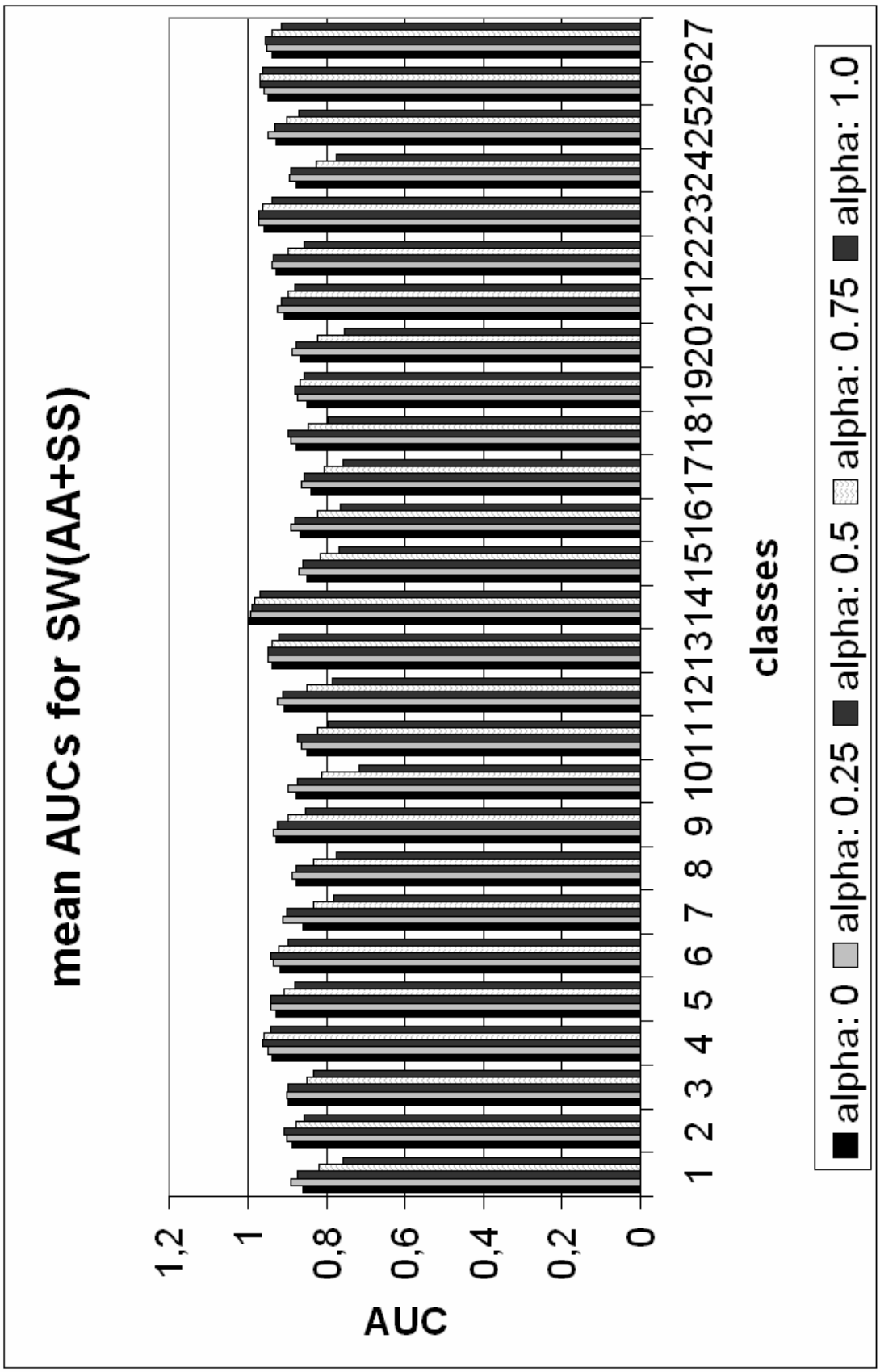

Figure 5.1: AUC values for SW0, SW25, SW50, SW75 and SW100

5.2. Compression-Based Classification

5.2.1 Classification using amino acid sequence and secondary structure on different levels

This section shows the experimental results obtained by using the NCD_β scores which is defined with Equation (3.9) in Section 3.2.1.1. The classification is made using the one-nearest-neighbor (1NN) classifier explained in Section 4.1.1. To include the secondary structure in different levels, β is set to 0, 0.25, 0.5, 0.75 and 1.0 and the classifiers obtained are called NCD0, NCD25, NCD50, NCD75 and NCD100 classifiers respectively. The classification results are shown in Table 5.4 and Figure 5.2.

Table 5.4: Mean AUC values for NCD0, NCD25, NCD50, NCD75 and NCD100 classifiers using 1NN

Class No	GO ID	NCD0	NCD25	NCD50	NCD75	NCD100
1	9405	0.76±0.03	0.74±0.02	0.71±0.03	0.70±0.04	0.69±0.04
2	9055	0.64±0.02	0.65±0.03	0.69±0.03	0.68±0.02	0.67±0.01
3	6810	0.63±0.02	0.65±0.02	0.67±0.03	0.62±0.03	0.62±0.02
4	16787	0.68±0.02	0.74±0.02	0.80±0.01	0.76±0.02	0.76±0.02
5	5506	0.67±0.02	0.73±0.13	0.70±0.03	0.72±0.03	0.73±0.02
6	166	0.63±0.03	0.68±0.02	0.74±0.02	0.74±0.01	0.73±0.01
7	3676	0.61±0.03	0.61±0.02	0.65±0.02	0.65±0.01	0.65±0.02
8	3700	0.63±0.02	0.67±0.03	0.74±0.01	0.65±0.02	0.65±0.02
9	6508	0.66±0.02	0.71±0.02	0.72±0.03	0.71±0.03	0.68±0.03
10	6412	0.65±0.03	0.65±0.03	0.69±0.03	0.70±0.03	0.67±0.03
11	3723	0.57±0.02	0.57±0.02	0.58±0.02	0.58±0.02	0.57±0.02
12	8270	0.62±0.02	0.66±0.02	0.68±0.02	0.66±0.02	0.66±0.03
13	5975	0.73±0.02	0.89±0.01	0.78±0.02	0.77±0.02	0.77±0.01
14	5179	0.99±0.00	0.99±0.00	0.99±0.00	0.99±0.00	0.99±0.00
15	16020	0.56±0.02	0.62±0.02	0.66±0.02	0.64±0.02	0.61±0.01
16	5515	0.58±0.02	0.60±0.02	0.62±0.02	0.63±0.02	0.61±0.02
17	5634	0.57±0.02	0.64±0.02	0.63±0.03	0.66±0.02	0.65±0.02
18	6355	0.62±0.02	0.66±0.01	0.65±0.02	0.66±0.02	0.66±0.01
19	5737	0.62±0.02	0.68±0.02	0.69±0.02	0.71±0.02	0.68±0.02
20	5622	0.56±0.02	0.60±0.01	0.61±0.01	0.62±0.01	0.60±0.02
21	5524	0.66±0.01	0.69±0.01	0.73±0.01	0.72±0.01	0.72±0.01
22	6118	0.61±0.00	0.66±0.01	0.69±0.01	0.69±0.01	0.68±0.01
23	16491	0.68±0.01	0.75±0.01	0.80±0.01	0.79±0.01	0.77±0.01
24	3677	0.57±0.01	0.61±0.02	0.61±0.02	0.60±0.02	0.59±0.01
25	5576	0.85±0.02	0.82±0.02	0.81±0.02	0.80±0.02	0.82±0.02
26	8152	0.70±0.01	0.77±0.01	0.82±0.01	0.82±0.01	0.82±0.01
27	3824	0.71±0.01	0.75±0.00	0.78±0.01	0.78±0.01	0.78±0.01
mean		0.66	0.69	0.71	0.71	0.70

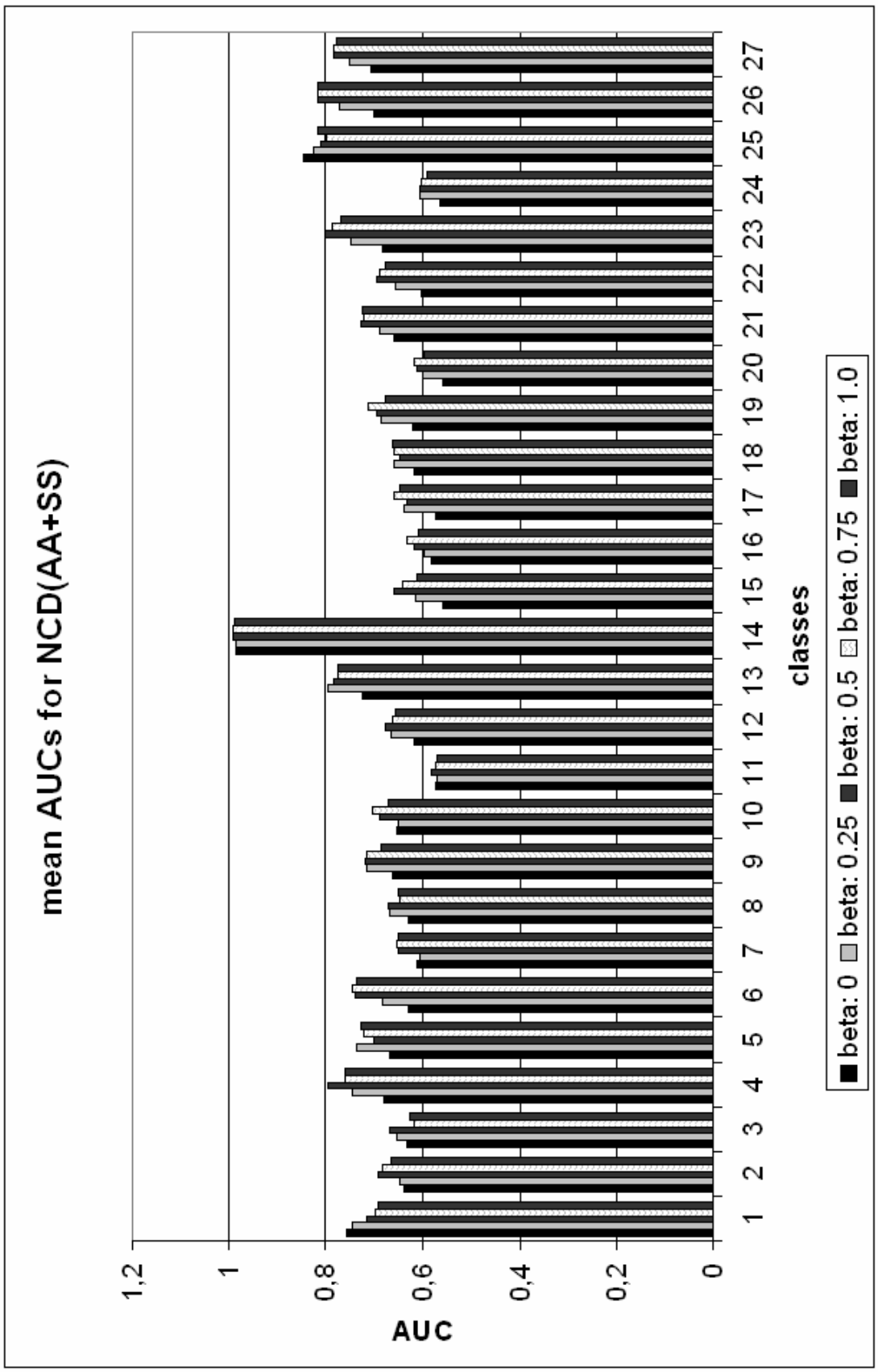

Figure 5.2: Mean AUC values for NCD0, NCD25, NCD50, NCD75 and NCD100

The best mean AUC value, which is 0.71, is obtained for NCD50 and NCD75 where β=0.5 and β=0.75 respectively. This followed by NCD100 with a mean AUC of 0.70 and by NCD25 with a mean AUC of 0.69. The worst performance (mean AUC: 0.66) is obtained for NCD00, the classifier using NCD scores obtained only from amino acid sequence. Class 14 is again an outlier with almost AUC = 1.00.

5.2.2 Classification using the joint representation

This section shows the experimental results obtained by using the NCD_{60} scores which is explained in Section 3.2.1.1 and the 1NN classifier explained in Section 4.1.1. The classification results are shown in Table 5.5. The mean AUC obtained by using the NCD_{60} scores is 0.69. Class 14 is again an outlier with almost AUC = 0.99.

Table 5.5: Mean AUC values using the NCD_{60} scores and the 1NN algorithm

Class No	GO ID	NCD60
1	9405	0.72 ± 0.04
2	9055	0.66 ± 0.03
3	6810	0.66 ± 0.03
4	16787	0.71 ± 0.02
5	5506	0.73 ± 0.03
6	166	0.71 ± 0.02
7	3676	0.59 ± 0.03
8	3700	0.70 ± 0.02
9	6508	0.67 ± 0.02
10	6412	0.60 ± 0.02
11	3723	0.59 ± 0.02
12	8270	0.61 ± 0.04
13	5975	0.78 ± 0.02
14	5179	0.99 ± 0.00
15	16020	0.61 ± 0.01
16	5515	0.61 ± 0.02
17	5634	0.65 ± 0.02
18	6355	0.68 ± 0.02
19	5737	0.67 ± 0.02
20	5622	0.58 ± 0.01
21	5524	0.69 ± 0.02
22	6118	0.67 ± 0.02
23	16491	0.75 ± 0.01
24	3677	0.61 ± 0.01
25	5576	0.83 ± 0.01
26	8152	0.75 ± 0.01
27	3824	0.73 ± 0.01
mean		0.69

5.3. Classification Using the Combined Similarity Metric

5.3.1 Classification using amino acid sequence and secondary structure on different levels

This section shows the experimental results obtained by using the $F_{\alpha\beta}$ scores which is defined with Equation (3.13) in Section 3.3. The classification is made using the one-nearest-neighbor (1NN) classifier explained in Section 4.1.1. The results obtained in Section 5.1.2 show that alignment-based classification had the best performance mostly at $\alpha = 0.25$, and for some classes at $\alpha = 0$ and $\alpha = 0.5$ (see Figure 5.1). Section 5.2.1 shows that compression-based classification performed best at $\beta = 0.5$ and $\beta = 0.75$ and for some particular classes at $\beta = 0$ and $\beta = 0.75$ (see Figure 5.2). Therefore, $F_{\alpha\beta}$ is tested for $\alpha = 0, 0.25, 0.5, 0.75$ and $\beta = 0, 0.25, 0.5, 0.75$ and 1.0. The classifiers obtained are called as in Table 5.6. The classification results are shown in Table 5.7.

Table 5.6: Classifier names for varying α and β values

	B				
α	**0**	**0.25**	**0.5**	**0.75**	**1.0**
0	F0_0	F0_25	F0_50	F0_75	F0_100
0.25	F25_0	F25_25	F25_50	F25_75	F25_100
0.5	F50_0	F50_25	F50_50	F50_75	F50_100
0.75	F75_0	F75_25	F75_50	F75_75	F75_100

At any value of α, worst performance is obtained at $\beta = 0$ and the mean AUC values obtained at any $\beta > 0$ are very close to each other. Ignoring the results at $\beta = 0$, the mean AUC value at $\alpha = 0$ and $\alpha = 0.25$ is 0.71 and the mean AUC value at $\alpha = 0.5$ is 0.72. Class 14 is again an outlier with the mean AUC very close to 1.00.

Classification made using the F_{δ} scores defined in Equation (3.15) in Section 3.3. The classification is made using the one-nearest-neighbor (1NN) classifier explained in Section 4.1.1. The parameter δ which controls the contribution of secondary structure is changed within the interval [0.0, 4.00] and α and β are set to 0.25 and 0.5, respectively, based on the results in Sections 5.1.2 and 5.2.1. The classification results are shown in Table 5.8.

Table 5.7: Mean AUC values using the $F_{\alpha\beta}$ scores and the 1NN algorithm

Class No	F0_0	F0_25	F0_50	F0_75	F0_100
1	0.71 ± 0.02	0.72 ± 0.02	0.71 ± 0.03	0.71 ± 0.03	0.72 ± 0.04
2	0.71 ± 0.03	0.74 ± 0.03	0.75 ± 0.02	0.72 ± 0.02	0.72 ± 0.02
3	0.69 ± 0.03	0.71 ± 0.02	0.67 ± 0.02	0.66 ± 0.03	0.68 ± 0.03
4	0.69 ± 0.02	0.72 ± 0.02	0.76 ± 0.02	0.77 ± 0.02	0.78 ± 0.02
5	0.72 ± 0.03	0.74 ± 0.03	0.73 ± 0.04	0.72 ± 0.03	0.72 ± 0.03
6	0.72 ± 0.02	0.74 ± 0.02	0.75 ± 0.02	0.76 ± 0.02	0.75 ± 0.02
7	0.60 ± 0.02	0.59 ± 0.02	0.59 ± 0.02	0.60 ± 0.02	0.61 ± 0.02
8	0.65 ± 0.02	0.69 ± 0.03	0.70 ± 0.03	0.68 ± 0.03	0.68 ± 0.03
9	0.70 ± 0.01	0.72 ± 0.02	0.73 ± 0.02	0.74 ± 0.02	0.74 ± 0.02
10	0.64 ± 0.03	0.61 ± 0.03	0.59 ± 0.03	0.58 ± 0.04	0.57 ± 0.04
11	0.58 ± 0.03	0.58 ± 0.02	0.60 ± 0.02	0.59 ± 0.02	0.57 ± 0.02
12	0.66 ± 0.03	0.66 ± 0.03	0.68 ± 0.03	0.69 ± 0.03	0.68 ± 0.03
13	0.77 ± 0.02	0.78 ± 0.01	0.79 ± 0.01	0.79 ± 0.02	0.79 ± 0.02
14	0.98 ± 0.01	0.99 ± 0.01	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00
15	0.59 ± 0.01	0.62 ± 0.01	0.62 ± 0.02	0.62 ± 0.02	0.61 ± 0.01
16	0.62 ± 0.02	0.59 ± 0.02	0.56 ± 0.02	0.58 ± 0.02	0.59 ± 0.02
17	0.62 ± 0.02	0.65 ± 0.02	0.66 ± 0.02	0.67 ± 0.02	0.66 ± 0.02
18	0.68 ± 0.02	0.69 ± 0.01	0.68 ± 0.01	0.65 ± 0.01	0.64 ± 0.01
19	0.63 ± 0.02	0.65 ± 0.02	0.69 ± 0.02	0.70 ± 0.02	0.69 ± 0.02
20	0.61 ± 0.02	0.62 ± 0.01	0.63 ± 0.02	0.61 ± 0.02	0.60 ± 0.02
21	0.71 ± 0.02	0.72 ± 0.02	0.75 ± 0.01	0.74 ± 0.01	0.74 ± 0.01
22	0.66 ± 0.01	0.68 ± 0.01	0.70 ± 0.02	0.67 ± 0.01	0.67 ± 0.01
23	0.76 ± 0.01	0.80 ± 0.01	0.81 ± 0.01	0.80 ± 0.01	0.80 ± 0.01
24	0.63 ± 0.01	0.63 ± 0.01	0.61 ± 0.02	0.60 ± 0.02	0.61 ± 0.01
25	0.87 ± 0.01	0.88 ± 0.01	0.84 ± 0.01	0.82 ± 0.02	0.81 ± 0.02
26	0.76 ± 0.01	0.80 ± 0.01	0.83 ± 0.01	0.84 ± 0.01	0.82 ± 0.01
27	0.78 ± 0.01	0.81 ± 0.01	0.82 ± 0.01	0.82 ± 0.01	0.83 ± 0.01
mean	0.69	0.71	0.71	0.71	0.71
Class No	**F25_0**	**F25_25**	**F25_50**	**F25_75**	**F25_100**
1	0.71 ± 0.02	0.71 ± 0.02	0.70 ± 0.03	0.71 ± 0.03	0.72 ± 0.04
2	0.71 ± 0.03	0.74 ± 0.03	0.75 ± 0.02	0.72 ± 0.02	0.72 ± 0.02
3	0.68 ± 0.02	0.69 ± 0.02	0.66 ± 0.02	0.66 ± 0.03	0.68 ± 0.03
4	0.70 ± 0.02	0.73 ± 0.02	0.77 ± 0.02	0.78 ± 0.02	0.79 ± 0.02
5	0.72 ± 0.03	0.74 ± 0.03	0.73 ± 0.04	0.72 ± 0.03	0.72 ± 0.03
6	0.72 ± 0.02	0.74 ± 0.02	0.76 ± 0.02	0.77 ± 0.02	0.76 ± 0.02
7	0.61 ± 0.02	0.60 ± 0.02	0.60 ± 0.02	0.61 ± 0.02	0.61 ± 0.02
8	0.67 ± 0.01	0.70 ± 0.03	0.70 ± 0.03	0.69 ± 0.03	0.68 ± 0.03
9	0.70 ± 0.03	0.73 ± 0.02	0.74 ± 0.02	0.74 ± 0.02	0.74 ± 0.02
10	0.63 ± 0.02	0.60 ± 0.04	0.58 ± 0.04	0.57 ± 0.04	0.57 ± 0.04
11	0.58 ± 0.03	0.58 ± 0.01	0.60 ± 0.02	0.58 ± 0.02	0.57 ± 0.02
12	0.65 ± 0.02	0.65 ± 0.03	0.68 ± 0.03	0.68 ± 0.03	0.68 ± 0.03
13	0.78 ± 0.01	0.78 ± 0.01	0.79 ± 0.01	0.79 ± 0.01	0.79 ± 0.01
14	0.98 ± 0.01	0.99 ± 0.01	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00

15	0.60 ± 0.02	0.63 ± 0.02	0.63 ± 0.02	0.63 ± 0.02	0.62 ± 0.02
16	0.60 ± 0.02	0.58 ± 0.02	0.56 ± 0.02	0.58 ± 0.02	0.58 ± 0.02
17	0.62 ± 0.02	0.64 ± 0.02	0.64 ± 0.02	0.66 ± 0.02	0.66 ± 0.02
18	0.67 ± 0.02	0.67 ± 0.01	0.67 ± 0.01	0.65 ± 0.01	0.64 ± 0.01
19	0.64 ± 0.01	0.66 ± 0.02	0.70 ± 0.02	0.71 ± 0.02	0.70 ± 0.02
20	0.59 ± 0.02	0.60 ± 0.01	0.62 ± 0.02	0.61 ± 0.02	0.60 ± 0.02
21	0.72 ± 0.02	0.73 ± 0.02	0.75 ± 0.01	0.75 ± 0.01	0.75 ± 0.01
22	0.66 ± 0.01	0.67 ± 0.01	0.70 ± 0.02	0.67 ± 0.02	0.67 ± 0.02
23	0.76 ± 0.01	0.80 ± 0.01	0.82 ± 0.01	0.80 ± 0.01	0.80 ± 0.01
24	0.62 ± 0.01	0.62 ± 0.01	0.60 ± 0.02	0.60 ± 0.02	0.61 ± 0.02
25	0.86 ± 0.01	0.86 ± 0.02	0.83 ± 0.01	0.82 ± 0.02	0.81 ± 0.02
26	0.77 ± 0.01	0.81 ± 0.01	0.84 ± 0.01	0.84 ± 0.01	0.83 ± 0.01
27	0.79 ± 0.01	0.81 ± 0.01	0.83 ± 0.01	0.83 ± 0.01	0.83 ± 0.01
mean	0.69	0.71	0.71	0.71	0.71
Class No	**F50_0**	**F50_25**	**F50_50**	**F50_75**	**F50_100**
1	0.71 ± 0.02	0.70 ± 0.02	0.70 ± 0.02	0.71 ± 0.03	0.72 ± 0.04
2	0.71 ± 0.03	0.74 ± 0.03	0.76 ± 0.03	0.73 ± 0.02	0.73 ± 0.02
3	0.68 ± 0.02	0.68 ± 0.03	0.66 ± 0.02	0.67 ± 0.03	0.69 ± 0.03
4	0.73 ± 0.02	0.76 ± 0.02	0.78 ± 0.02	0.79 ± 0.02	0.79 ± 0.02
5	0.73 ± 0.03	0.75 ± 0.03	0.74 ± 0.03	0.72 ± 0.03	0.72 ± 0.03
6	0.75 ± 0.02	0.76 ± 0.02	0.77 ± 0.02	0.78 ± 0.02	0.77 ± 0.02
7	0.64 ± 0.02	0.63 ± 0.03	0.62 ± 0.02	0.63 ± 0.02	0.63 ± 0.02
8	0.70 ± 0.02	0.71 ± 0.02	0.72 ± 0.02	0.70 ± 0.02	0.69 ± 0.03
9	0.72 ± 0.02	0.74 ± 0.02	0.75 ± 0.02	0.75 ± 0.02	0.75 ± 0.02
10	0.60 ± 0.04	0.61 ± 0.04	0.60 ± 0.04	0.58 ± 0.04	0.58 ± 0.04
11	0.56 ± 0.02	0.58 ± 0.01	0.60 ± 0.01	0.60 ± 0.01	0.59 ± 0.02
12	0.66 ± 0.03	0.66 ± 0.03	0.69 ± 0.03	0.69 ± 0.03	0.69 ± 0.03
13	0.79 ± 0.02	0.79 ± 0.02	0.80 ± 0.01	0.79 ± 0.01	0.79 ± 0.01
14	0.99 ± 0.01	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00
15	0.60 ± 0.02	0.62 ± 0.02	0.62 ± 0.02	0.63 ± 0.02	0.63 ± 0.02
16	0.61 ± 0.02	0.60 ± 0.02	0.57 ± 0.02	0.59 ± 0.02	0.59 ± 0.03
17	0.64 ± 0.02	0.66 ± 0.02	0.65 ± 0.02	0.66 ± 0.02	0.66 ± 0.02
18	0.69 ± 0.01	0.68 ± 0.01	0.68 ± 0.01	0.66 ± 0.01	0.65 ± 0.01
19	0.67 ± 0.02	0.68 ± 0.02	0.71 ± 0.02	0.72 ± 0.02	0.70 ± 0.02
20	0.58 ± 0.02	0.60 ± 0.01	0.62 ± 0.02	0.61 ± 0.02	0.60 ± 0.02
21	0.73 ± 0.02	0.74 ± 0.02	0.76 ± 0.01	0.76 ± 0.01	0.76 ± 0.01
22	0.68 ± 0.02	0.69 ± 0.02	0.71 ± 0.02	0.68 ± 0.02	0.68 ± 0.01
23	0.79 ± 0.01	0.81 ± 0.01	0.83 ± 0.01	0.82 ± 0.01	0.81 ± 0.01
24	0.64 ± 0.01	0.63 ± 0.01	0.61 ± 0.02	0.62 ± 0.02	0.63 ± 0.02
25	0.85 ± 0.01	0.86 ± 0.02	0.84 ± 0.01	0.82 ± 0.02	0.82 ± 0.02
26	0.80 ± 0.01	0.83 ± 0.01	0.86 ± 0.01	0.86 ± 0.01	0.84 ± 0.01
27	0.81 ± 0.01	0.82 ± 0.01	0.84 ± 0.01	0.84 ± 0.01	0.84 ± 0.01
mean	0.71	0.72	0.72	0.72	0.72

The mean AUC values obtained with δ regulation start with 0.76 at δ = 0.25 and increase up to 0.78 at δ = 2, then it starts decreasing for δ > 2. Class 14 is again an outlier for all values of δ with the mean AUC very close to 1.00.

Table 5.8: Mean AUC values using the F_δ scores and the 1NN algorithm

Class No	GO ID	δ					
		0.25	**0.50**	**0.75**	**2.00**	**3.00**	**4.00**
1	9405	0.80 ± 0.02	0.80 ± 0.02	0.80 ± 0.02	0.82 ± 0.02	0.80 ± 0.02	0.80 ± 0.02
2	9055	0.77 ± 0.02	0.77 ± 0.02	0.77 ± 0.02	0.76 ± 0.03	0.76 ± 0.02	0.76 ± 0.02
3	6810	0.73 ± 0.04	0.73 ± 0.03	0.73 ± 0.03	0.75 ± 0.03	0.74 ± 0.02	0.73 ± 0.02
4	16787	0.82 ± 0.02	0.83 ± 0.02	0.83 ± 0.02	0.83 ± 0.01	0.83 ± 0.01	0.83 ± 0.01
5	5506	0.75 ± 0.02	0.76 ± 0.02	0.77 ± 0.02	0.80 ± 0.03	0.78 ± 0.02	0.78 ± 0.02
6	166	0.81 ± 0.01	0.81 ± 0.01	0.81 ± 0.02	0.80 ± 0.02	0.81 ± 0.01	0.81 ± 0.01
7	3676	0.67 ± 0.02	0.67 ± 0.02	0.69 ± 0.02	0.70 ± 0.02	0.69 ± 0.02	0.69 ± 0.02
8	3700	0.72 ± 0.02	0.71 ± 0.02	0.72 ± 0.02	0.76 ± 0.02	0.72 ± 0.02	0.72 ± 0.02
9	6508	0.77 ± 0.02	0.77 ± 0.01	0.78 ± 0.02	0.79 ± 0.02	0.78 ± 0.02	0.78 ± 0.02
10	6412	0.69 ± 0.04	0.76 ± 0.04	0.77 ± 0.04	0.76 ± 0.02	0.77 ± 0.03	0.77 ± 0.03
11	3723	0.63 ± 0.02	0.66 ± 0.02	0.67 ± 0.02	0.67 ± 0.01	0.67 ± 0.02	0.66 ± 0.02
12	8270	0.75 ± 0.03	0.77 ± 0.03	0.78 ± 0.03	0.77 ± 0.02	0.77 ± 0.03	0.77 ± 0.03
13	5975	0.83 ± 0.02	0.85 ± 0.02	0.85 ± 0.02	0.85 ± 0.01	0.84 ± 0.02	0.84 ± 0.02
14	5179	0.99 ± 0.00	0.99 ± 0.00	1.00 ± 0.00	0.99 ± 0.001	1.00 ± 0.00	1.00 ± 0.00
15	16020	0.69 ± 0.02	0.70 ± 0.01	0.72 ± 0.01	0.73 ± 0.03	0.74 ± 0.02	0.74 ± 0.02
16	5515	0.65 ± 0.02	0.66 ± 0.02	0.67 ± 0.03	0.69 ± 0.02	0.68 ± 0.02	0.68 ± 0.02
17	5634	0.69 ± 0.02	0.68 ± 0.02	0.68 ± 0.03	0.69 ± 0.01	0.67 ± 0.03	0.67 ± 0.03
18	6355	0.72 ± 0.01	0.72 ± 0.01	0.71 ± 0.01	0.75 ± 0.02	0.72 ± 0.02	0.72 ± 0.02
19	5737	0.75 ± 0.02	0.75 ± 0.01	0.76 ± 0.02	0.75 ± 0.01	0.75 ± 0.02	0.75 ± 0.02
20	5622	0.68 ± 0.02	0.70 ± 0.02	0.70 ± 0.02	0.68 ± 0.01	0.70 ± 0.02	0.70 ± 0.02
21	5524	0.79 ± 0.01	0.79 ± 0.01	0.80 ± 0.01	0.79 ± 0.01	0.79 ± 0.01	0.79 ± 0.01
22	6118	0.76 ± 0.01	0.78 ± 0.01	0.78 ± 0.01	0.80 ± 0.01	0.79 ± 0.02	0.79 ± 0.02
23	16491	0.86 ± 0.01	0.87 ± 0.01	0.87 ± 0.01	0.85 ± 0.01	0.86 ± 0.01	0.86 ± 0.01
24	3677	0.66 ± 0.02	0.67 ± 0.02	0.68 ± 0.01	0.70 ± 0.01	0.68 ± 0.02	0.68 ± 0.02
25	5576	0.86 ± 0.01	0.86 ± 0.02	0.86 ± 0.02	0.88 ± 0.01	0.87 ± 0.02	0.86 ± 0.02
26	8152	0.87 ± 0.01	0.87 ± 0.01	0.87 ± 0.01	0.87 ± 0.01	0.87 ± 0.01	0.87 ± 0.01
27	3824	0.84 ± 0.01	0.84 ± 0.01	0.84 ± 0.01	0.83 ± 0.01	0.83 ± 0.01	0.83 ± 0.01
mean		0.76	0.77	0.77	0.78	0.77	0.77

5.3.2 Classification using the joint representation

This section shows the experimental results obtained by using the F_{60} scores which is defined with Equation (3.16) in Section 3.3. The classification is made using the one-nearest-neighbor (1NN) classifier explained in Section 4.1.1. The mean AUC value obtained is 0.71 (see Table 5.9) which is 0.20 higher than the mean AUC values obtained with classifying done with NCD_{60} scores, joint representation not including

the Smith-Waterman alignment scores (see Table 5.5). Class 14 is again an outlier with the mean AUC very close to 1.00.

Table 5.9: Mean AUC values using the F_{60} scores and the 1NN algorithm

Class No	GO ID	F_{60}
1	9405	0.72 ± 0.02
2	9055	0.70 ± 0.03
3	6810	0.66 ± 0.02
4	16787	0.73 ± 0.03
5	5506	0.74 ± 0.03
6	166	0.75 ± 0.03
7	3676	0.64 ± 0.03
8	3700	0.71 ± 0.02
9	6508	0.74 ± 0.02
10	6412	0.58 ± 0.03
11	3723	0.58 ± 0.01
12	8270	0.68 ± 0.04
13	5975	0.79 ± 0.02
14	5179	0.99 ± 0.00
15	16020	0.60 ± 0.02
16	5515	0.56 ± 0.02
17	5634	0.64 ± 0.03
18	6355	0.67 ± 0.01
19	5737	0.68 ± 0.02
20	5622	0.58 ± 0.02
21	5524	0.77 ± 0.01
22	6118	0.67 ± 0.02
23	16491	0.80 ± 0.01
24	3677	0.64 ± 0.01
25	5576	0.86 ± 0.02
26	8152	0.81 ± 0.01
27	3824	0.80 ± 0.00
mean		0.71

5.3.3 Classification using all features

This section shows the experimental results obtained by the F_{ALL} classifiers described with Figure 3.7 in Section 3.4. The classification is made using the one-nearest-neighbor (1NN) classifier explained in Section 4.1.1. The mean AUC value obtained is 0.65 (see Table 5.10). Class 14 is again an outlier for all values of δ with the mean AUC very close to 1.00.

Table 5.10: Mean AUC values using the F_{ALL} feature vector and the 1NN algorithm

Class No	GO ID	F_{ALL}
1	9405	0.70 ± 0.03
2	9055	0.71 ± 0.02
3	6810	0.61 ± 0.02
4	16787	0.65 ± 0.02
5	5506	0.60 ± 0.03
6	166	0.64 ± 0.03
7	3676	0.67 ± 0.01
8	3700	0.53 ± 0.02
9	6508	0.60 ± 0.02
10	6412	0.59 ± 0.03
11	3723	0.56 ± 0.02
12	8270	0.71 ± 0.02
13	5975	0.72 ± 0.02
14	5179	0.96 ± 0.01
15	16020	0.63 ± 0.03
16	5515	0.62 ± 0.03
17	5634	0.57 ± 0.01
18	6355	0.58 ± 0.02
19	5737	0.64 ± 0.01
20	5622	0.57 ± 0.02
21	5524	0.68 ± 0.01
22	6118	0.72 ± 0.01
23	16491	0.80 ± 0.01
24	3677	0.55 ± 0.01
25	5576	0.82 ± 0.02
26	8152	0.68 ± 0.02
27	3824	0.57 ± 0.01
mean		0.65

6. CONCLUSIONS AND FUTURE WORK

In this study, it is found out that using the whole amino acid sequences, as opposed to portions belonging to different secondary structure elements, results in the best function prediction performance. Using HL regions together results in almost as good performance as the whole sequence. On the other hand, E regions are the least significant in function prediction. When learning only from positive examples (tNN), HE follows the performance of HL and the distribution of H, E and L does not play a significant role. However, using kNN algorithm which takes into account both positive and negative examples produces better prediction results.

Figure 6.1 compares the AUC values for the classification SW_{AA} and NCD_{AA} using the 1NN classification algorithm. As expected, the alignment-based classification has a better performance than the compression-based classification when using amino acid sequence only.

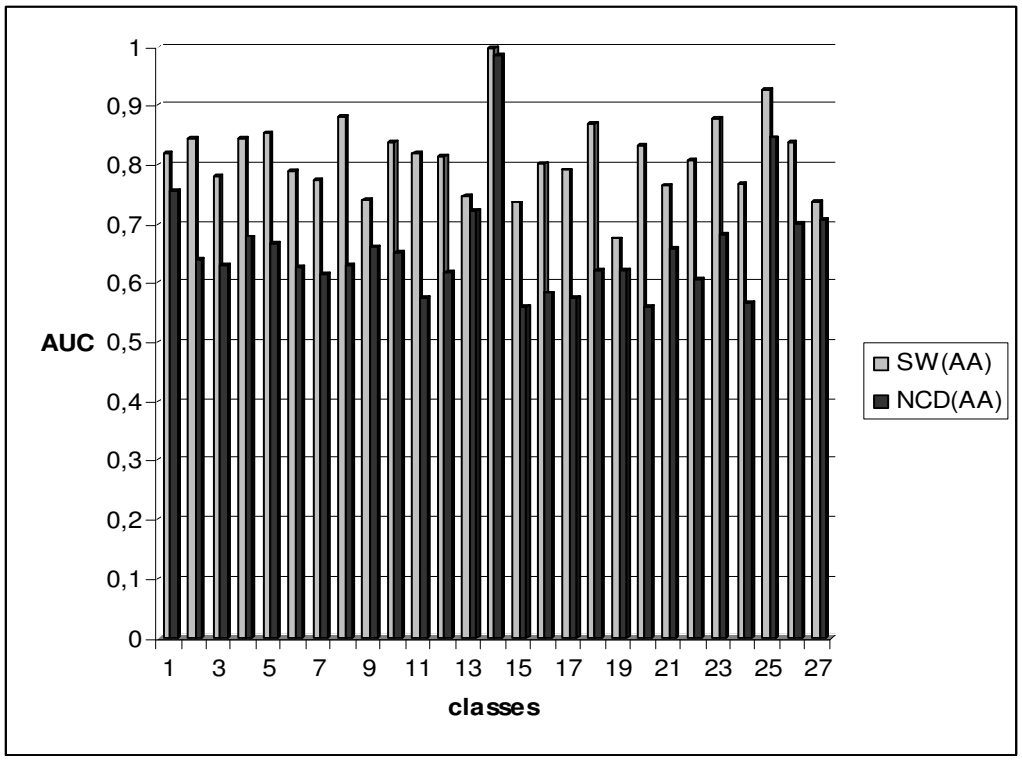

Figure 6.1: Comparison of AUC values of SW_{AA} and NCD_{AA} using 1NN

Including the secondary structure to both Smith-Waterman and NCD scores leads to a better classification performance. The best performance for alignment-based classification is obtained at $\alpha = 0.25$ and for compression-based classification at $\beta = 0.50$ (see Figure 6.2).

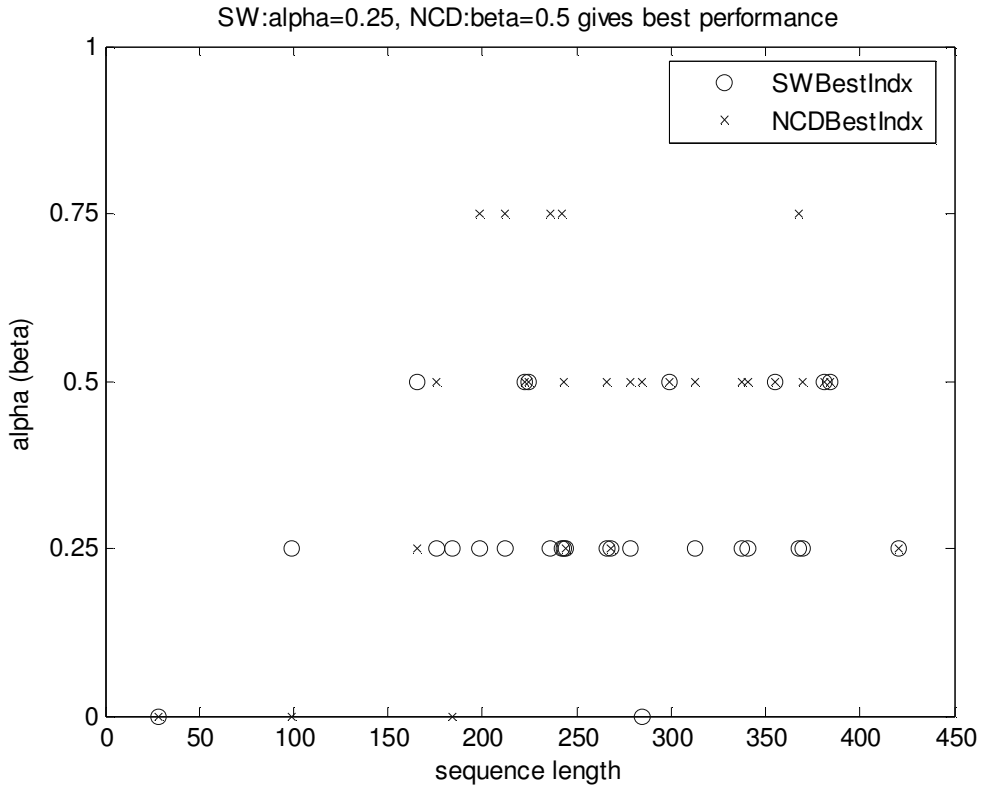

Figure 6.2: α and β values at which best classification performance is obtained

Whereas Smith-Waterman-only classification performs best at $\alpha = 0.25$, the combined metric has the best performance with the mean AUC 0.72 at $\alpha = 0.50$. For the combined metric, the results obtained at $\beta = 0.25$, 0.5 and 0.75 are very close and all three of them are higher than the results obtained at $\beta = 0$ (see Table 5.7). So, it can be concluded that classifiers using the combined metric that incorporates the secondary structure in the Smith-Waterman scores at 50% and in NCD scores at 25-75%.

When the contribution of the NCD scores is increased using the δ parameter, the mean AUC values increase up to 0.78 while $\delta \leq 2$ and decrease for $\delta > 2$. Therefore it can be concluded that using a similarity score combined of Smith-Waterman and NCD scores, both including the amino acid sequence and the secondary structure, can be obtained with $\delta = 2$.

Including the Smith-Waterman scores into the NCD scores computed using the joint representation improves the classification performance, however these scores, as well using Smith-Waterman and NCD scores together, are outperformed by the previously reported combined similarity metric.

Additional interesting methods to combine the alignment scores with the normalized compression distance scores are also possible. Amino acids and secondary structure elements having low substitution costs according to the substitution matrix used in the alignment algorithm can be represented with the same symbol by implementing a penalty proportional to the substitution cost and a more robust joint representation can be produced. Besides, the substitution matrix itself can be involved in the compression algorithm, especially in building up the compression dictionary. These approaches are not tested in this study and considered as future work.

REFERENCES

7-zip.org, n.d. The LZMA algorithm, *[Online],* Available from: http://www.7-zip.org/sdk.html

Alpaydın, E., 2004. Introduction to Machine Learning, The MIT Press, Massachusetts.

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

Ashburner, M., **1998.** On the representation of gene function in genetic databases, *Proceedings of the Intelligent Systems for Molecular Biology*, Montreal, **6**.

Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S. and Eppig, J., 2000. Gene Ontology: tool for the unification of biology, *Nat. Genet.*, **25**, 25–29.

Aygün, E., Kömürlü, C., Aydın, Z. and Çataltepe, Z., 2008. Protein Function Prediction with Amino Acid Sequence and Secondary Structure Alignment Scores, *HIBIT 2008*, Istanbul, Turkey, May 18-20.

Aygün, E., and Çataltepe, Z., 2008. balign, *in preparation*.

Bennett, C.H., Gacs, P., Li, M., Vitanyi, P.M.B. and Zurek, W., 1998. Information distance, *IEEE Trans. Inf. Theory*, **44(4)**, 1407–1423.

Benedetto, D., Caglioti, E. and Loreto, V., 2002. Language trees and zipping. *Phys. Review Lett.*, *88(4)*.

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. and Bourne, P., 2000. The Protein Data Bank, *Nucleic Acids Research*, **28(1),** 235-242.

Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.C., Estreicher, A.,Gasteiger, E., Martin, M.J., Michoud, K., O'Donovan, C., Phan, I. *et al*., 2003 The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003, *Nucleic Acids Res.*, **31**, 365-370.

Bradley, A.P., 1997. The use of area under the ROC curve in the evaluation of machine learning algorithms, *Pattern Recognition*, **30(7)**, 1145-1159.

Bratko, A., and Filipic. B., 2005. Spam filtering using compression models. *Department of Intelligent Systems Technical Report*, **IJS-DP-9227**, Jozef Stefan Institute, Ljubljana, Slovenia.

Butler,D., 2002. NIH pledges cash for global protein database, *Nature,* **419**, 101.

Bzip.org, n.d. Bzip2, *[Online],* Available from: http://www.bzip.org/

Camon, E., Magrane, M., Barrell, D., Lee, V., Dimmer, E., Maslen, J., Binns, E., Harte, N., Lopez, R. and Apweiler, R., 2004. The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology, *Nucleic Acids Research*, **32(1)**, D262-D266.

Cheng, J. and Baldi, P., 2006. A machine learning information retrieval approach to protein fold recognition, *Bioinformatics*, **22(12)**, 1456–1463.

Cilibrasi, R., 2003. The CompLearn Toolkit, *(Online)*. Available: http://complearn.sourceforge.net/

Cilibrasi, R., Vitanyi, P.M.B. and Wolf, R., 2004. Algorithmic clustering of music based on string compression, *Computer Music J.,* **28(4)**, 49–67.

Cilibrasi, R. and Vitanyi, P.M.B., 2005. Clustering by compression, *IEEE Trans. Inform. Th.*, **51(4)**, 523–1545.

Cataltepe, Z., Yaslan, Y. and Sönmez, A., 2006. Music genre classification using midi and audio features, *Journal of Applied Signal Processing*, 2006.

Çataltepe, Z., Aygün, E., Filiz, A., Keskin, Ö, Kömürlü, C. and Altunbaşak, Y., 2007. Dimensionality Reduction for Protein Function Prediction, *Automated Function Prediction – Biosapiens Joint Special Interest Group Meeting*, Vienna, Austria.

Ferragina, P., Giancarlo, R., Greco, V., Manzini, G. and Valiente, G., 2007. Compression-based classification of biological sequences and structures via the Universal Similarity Metric: experimental assessment, *BMC Bioinformatics*, **8**, 252.

Filiz, A., Aygün, E., Keskin, Ö. and Çataltepe, Z., 2008. Importance of Secondary Structure Elements for Prediction of GO Annotations, *HIBIT 2008*, Istanbul, Turkey.

Freschi, V. and Bogliolo, A., 2005. Using sequence compression to speedup probabilistic profile matching, *Bioinformatics*, **21(10),** 2225-2229.

Gailly, J., n.d. Data Compression, *[Online]*, Available from: http://gailly.net/

The Gene Ontology Consortium (GO), 2000. Gene ontology tool for the unification of biology, *Nat. Genet.*, **25**, 25-29.

Grundy, N., 1998. Family-based homology detection via pair-wise sequence comparison, Proc. 2nd Ann. Int. Conf. Computational Molecular Biology, 94–100.

Gzip.org, n.d. *[Online],* Available from: http://www.gzip.org/

Hanley, **J.A. and McNeil, B. J., 1982.** The meaning and use of the area under a receiver operating characteristic (ROC) curve, *Radiology*, 143(1), 29-36.

Hategan, A. and Tabus, I., 2004. Protein is compressible, *Proceedings of the Northern Signal Processing Symposium*.

IUPAC-IUB Joint Commission on Biochemical Nomenclature, 1984. Nomenclature and Symbolism for Amino Acids and Peptides, *Eur. J. Biochem.*, **138,** 9-37.

Jaakkola, T., Diekhans, M. and Haussler, D., 1999. Using the Fisher kernel method to detect remote protein homologies, *Proc. 7th Int. Conf. Intelligent Systems for Molecular Biology*, 149–158.

Jaakkola, T., Diekhans, M. and Haussler, D., 2000. A discriminative framework for detecting remote protein homologies, *Journal of Computational Biology*, **7(1–2)**, 95–114.

Keogh, E., Lonardi, S. and Rtanamahatana, C.A., 2004. Toward parameter free data mining, *In Proceedings of the 10th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Seattle, WA, 206–215, Aug 22-25.

Kimball, J.W., 2008. J.W. Kimball's Biology Pages, *(Online)*, Available: http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/P/Peptide.gif, Access date: 26.04.2008.

Kleinberg, J. and Tardos, E., 2006. Algorithm Design, Pearson Education, Inc.

Kocsor, A., Kertesz-Farkas, A., Kajan, L. and Pongor, S., 2005. Application of compression-based distance measures to protein sequence classification: a methodological study, *Bioinformatics*, **22(4)**, 407–412.

Krasnogor, N. and Pelta, D.A., 2004. Measuring the similarity of protein structures by means of the universal similarity metric, *Bioinformatics,* **20(7)**, 1015-1021.

Krogh, A., Brown, M., Mian, I., Sjolander, K. and Haussler, D., 1994. Hidden Markov models in computational biology: Applications to protein modeling, *Journal of Molecular Biology*, **235**, 1501–1531.

Li, M. and Vitanyi, P.M.B., 1997. An Introduction to Kolmogorov complexity and its Applications. Springer Verlag, NY.

Li, M., Badger, J.H., Chen, X., Kwong, S., Kearney, P. and Zhang, H. , 2001. An information-based sequence distance and its application to whole mitochondrial genome phylogeny, *Bioinformatics*, **17**, 149–154.

Liao, L. and Noble, W.S., 2003. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships, *Journal of Comp. Biology*, **10(6)**, 857-868.

Nevill-Manning, C.G. and Witten, I.H., 1999. Protein is incompressible, *DCC '99 Data Compression Conference*, 257.

National Center for Biology Information (NCBI) , 2004. NCBI Glossary, *(Online),* Available: http://www.ncbi.nlm.nih.gov/Education/ BLASTinfo/glossary2.html , 2004.

Needleman, S. and Wunsch, C., 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins, *J Mol Biol.*, **48(3)**, 443-53.

National Center for Biology Information (NCBI) , 2004. Phylogenetics Factsheet, *(Online),* Available: http://www.ncbi.nlm.nih.gov/About/primer/ phylo.html, Access date: 27.04.2008.

Pauling, L., Corey. R.B. and Branson, H.R., 1951. The structure of proteins: Two hydrogen-bonded helical configurations of the polypeptide chain., *Proceedings of the National Academy of Sciences of Washington*, **37**, 205-211.

Pauling, L. and Corey. R.B., 1951. Configurations of polypeptide chains with favored orientations of the polypeptide around single bonds: Two pleated sheets, *Proceedings of the National Academy of Sciences of Washington*, **37**, 729-740.

Passerini, A., Punta, M., Ceroni, A., Rost B. and Frasconi, P., 2006. Identifying cysteines and histidines in transition-metal-binding sites using support vector machines and neural networks, *Proteins: Structure, Function, and Bioinforma-ics*, **65**, 305–316.

Plant and Soil Sciences e-Library, 2006. Inhibitors of Aromatic Amino Acid Biosynthesis, *(Online),* Available from: http://plantandsoil.unl.edu/croptechnology2005/UserFiles/Image/siteImages/AminoAcidLG.gif, Access date: 26.04.2008.

RCSB Protein Data Bank (PDB), nd. *[Online]*, Available from: http://www.rcsb.org/pdb/home/home.do

Sayood, K., 1996. Introduction to Data Compression, Morgan Kaufmann Publishers, Inc., San Fransisco, California.

Science College, nd. Peptides and Proteins, *(Online),* Available from: http://www.sciencecollege.co.uk/SC/biochemicals/bsheet.gif, Access date: 26.04.2008.

Sculley, D. and Brodley, C.E., 2006. Compression and Machine Learning: A New Perspective on Feature Space Vectors, *Proceedings of the Data Compression Conference (DCC'06)*.

Smith, T.F. and Waterman, M.S., 1981. Identification of Common Molecular Subsequences. *Journal of Molecular Biology,* **147,** 195-197.

Soeding, J., 2005. Protein homology detection by HMM-HMMcomparison, *Bioinformatics*, **21**, 951–960.

StatSoft, Inc., 2007. Electronic Statistics Textbook, Tulsa, OK, USA. *(Online)* Available from: http://www.statsoft.com/textbook/stathome.html, Access date: 26.04.2008.

University of Miami Department of Biology, nd. Proteins: Function and Structure, *(Online)*, Available from: http://fig.cox.miami.edu/~cmallery/150/protein/alpha-helix.jpg, Access date: 26.04.2008.

University of Guelf Department of Chemistry and Biochemistry, 2000. Biophysical Methods Lecture Notes, A tour of protein structure: Common folding patterns of protein tertiary structure, *(Online),* Available from: http://www.chembio.uoguelph.ca/educmat/phy456/protstr4.htm, Access date: 26.04.2008.

Wallqvist, A., Fukunishi, Y., Murphy, L.R., Fadel, A. and Levy R. M., 2000. Iterative sequence/secondary structure search for protein homologs: comparison with amino acid sequence alignments and application to

fold recognition in genome databases, *Bioinformatics,* **16(11)**, 988-1002.

Westbrook, J., Feng , Z., Chen, L., Yang, H. and Berman, H.M., 2003. The Protein Data Bank and structural genomics, *Nucleic Acids Research,* **31**, 489-491.

Wu, C.H., Huang, H., Yeh, L.S. and Barker, W.C., 2003. Protein family classification and functional annotation, *Comput. Biol. Chem.,* **27**, 37-47.

Yu, L. and Liu, H., 2004. Efficient feature selection via analysis of relevance and redundancy, *Journal of Machine Learning Research*, **5**, 1205-1224.

Ziv, J. and Lempel, A., 1977. A universal algorithm for data compression, *IEEE Transactions on Information Theory,* **23(3),** 337-343.

Ziv, J. and Lempel, A., 1978. Compression of individual sequences via variable-rate coding, *IEEE Transactions on Information Theory,* **24(5),** 530-536.

AUTOBIOGRAPHY

Aslı Filiz is born in 1983 in Izmir. In the year 2002, she graduated from German High School of Istanbul and started the undergraduate program for Computer Engineering at Istanbul Technical University during which she was supported with the Istanbul Technical University Rectorate's scholarship of merit. In 2006, she graduated with the degree of computer engineer. The graduation project under the supervision of Assoc. Prof. Dr. Feza Buzluca was about a web-based smart marketing system involving data mining. In the same year, she started the Master of Science program for Computer Science at Informatics Institute at Istanbul Technical University and joined the Bioinformatics Project coordinated by Assoc. Prof. Dr. Zehra Çataltepe from the Department of Computer Engineering at Istanbul Technical University. During the Master program, she was supported by TUBITAK's National Scholarship Program for MS Students (2228). Based on her work in the Bioinformatics Project, the poster "Dimensionality Reduction for Protein Function Prediction" is accepted Automated Function Prediction – Biosapiens Joint Special Interest Group Meeting in Austria in July 2007 and the paper titled "Importance of Secondary Structure Elements for Prediction of GO Annotations" is accepted to HIBIT 2008 for oral presentation. The paper titled "Gene Ontology Prediction Using Compression Based Distances and Alignment Scores on Both Amino Acid Sequence and Secondary Structure" is submitted to ISCIS 2008 and a journal paper is in preparation.