PREDICTION OF ENZYMATIC PROPERTIES OF PROTEIN SEQUENCES BASED ON THE ENZYME COMMISSION NOMENCLATURE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ALPEREN DALKIRAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2017

Approval of the thesis:

PREDICTION OF ENZYMATIC PROPERTIES OF PROTEIN SEQUENCES BASED ON THE ENZYME COMMISSION NOMENCLATURE

submitted by **ALPEREN DALKIRAN** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering** ______________

Prof. Dr. M. Volkan Atalay
Supervisor, **Computer Engineering Department, METU** ______________

Prof. Dr. Rengül Çetin-Atalay
Co-supervisor, **Graduate School of Informatics, METU** ______________

Examining Committee Members:

Prof. Dr. Hasan Oğul
Computer Engineering Department, Başkent University ______________

Prof. Dr. M. Volkan Atalay
Computer Engineering Department, METU ______________

Assoc. Prof. Dr. Pınar Karagöz
Computer Engineering Department, METU ______________

Assoc. Prof. Dr. Sinan Kalkan
Computer Engineering Department, METU ______________

Assist. Prof. Dr. Nurcan Tunçbağ
Graduate School of Informatics, METU ______________

Date: ______________

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

Name, Last Name: ALPEREN DALKIRAN

Signature :

ABSTRACT

PREDICTION OF ENZYMATIC PROPERTIES OF PROTEIN SEQUENCES
BASED ON THE ENZYME COMMISSION NOMENCLATURE

Dalkıran, Alperen
M.S., Department of Computer Engineering
Supervisor : Prof. Dr. M. Volkan Atalay
Co-Supervisor : Prof. Dr. Rengül Çetin-Atalay

September 2017, 74 pages

The volume of expert manual annotation of biomolecules is steady due to high costs associated with it, although the number of sequenced genomes continues to grow exponentially. Computational methods have been proposed in order to predict the attributes of gene products. The prediction of Enzyme Commission (EC) numbers is a challenging issue in this area. Enzymes have crucial roles in metabolic pathways, therefore they are widely employed in biotechnological and biomedical applications. EC numbers are numerical representations of enzymatic functions based on chemical reactions that they catalyze. Due to the cost and labor extensiveness of in vitro experiments EC classification annotation of catalytically active proteins are limited. Therefore, computational tools have been proposed to classify these proteins to annotate them with EC nomenclature. However, the performance of existing tools indicates that EC number prediction still requires improvement. Here, we present an EC number prediction tool, *ECPred*, to obtain predictions for large-scale protein sets. In *ECPred*, we employed hierarchical data preparation and evaluation steps by utilizing the functional relations among the four levels of EC annotation system. The main features that distinguish our approach from existing studies are the use of a combination of independent classifiers, and novel data preparation and evaluation methods. Totally, 858 EC classifiers are trained which consists of 6 main, 55 subfamily, 163 sub-subfamily and 634 substrate EC class classifiers. The average F-score value of 0.99 is obtained for all EC classes using the validation datasets. Enzyme or non-enzyme classification is incorporated into *ECPred* along with a hierarchical prediction approach. To the best of our knowledge, this is the first study that predicts the enzymatic function of proteins starting from Level 0 (enzyme/non-enzyme) going up to Level 4 (substrate class). Finally, *ECPred* is compared with other similar tools on independent test sets and *ECPred* obtained better results than existing tools, however, the results show that there is still room for improvement.

Keywords: Enzyme, Enzyme Commision Number, Machine Learning, Sequence Analysis

ÖZ

PROTEİN SEKANSLARININ ENZİMATİK ÖZELLİKLERİNİN ENZİM KOMİSYONU TERMİNOLOJİSİNE DAYALI TAHMİNİ

Dalkıran, Alperen
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. M. Volkan Atalay
Ortak Tez Yöneticisi : Prof. Dr. Rengül Çetin-Atalay

Eylül 2017 , 74 sayfa

Sekanslanan gen sayısı gün geçtikçe katlanarak artmaya devam ederken uzman yardımıyla anlamlandırılan biyomolekül bu işlemin yüksek maliyet gerektirmesinden dolayı sınırlı sayıda kalmaktadır. Gen ürünlerinin özelliklerini tahmin etmek için algoritmaya dayalı yöntemler literatürde önerilmiştir. Enzim Komisyonu (EC) numaralarının tahmini bu alandaki zor bir konudur. Enzimler metabolik yolaklarda önemli rol oynamaktadır ve bu nedenle biyoteknoloji ve biyomedikal uygulamalarında yaygın olarak kullanılmaktadırlar. EC numaraları, katalize ettikleri kimyasal reaksiyonlara dayalı enzimatik fonksiyonların sayısal temsilidir. Laboratuvar ortamında yapılan deneylerin maliyetinin yüksekliği ve çok fazla işgücü gerektirmesinden ötürü, katalik olarak aktif olan proteinlerin EC sınıflandırması ile anlamlandırılması sınırlıdır. Bu nedenle, bu proteinleri EC terminolojisiyle sınıflandırıp anlamlandırmak için algoritmaya dayalı yöntemler önerilmiştir. Bununla birlikte, mevcut araçların performans sonuçları, EC numarası tahmin alanının hala iyileştirilmesi gerektiğini göstermektedir. Bu çalışmada, büyük ölçekli protein kümeleri için tahminler elde etmek için EC numarası tahmini yapan bir araç, *ECPred* anlatılmaktadır. *ECPred*'de, dört seviyeli EC anlamladırma sistemi arasındaki işlevsel ilişkileri kullanarak hiyerarşik veri hazırlama ve değerlendirme aşamaları geliştirildi. Yaklaşımımızı mevcut çalışmalardan ayıran başlıca özellikler, bağımsız sınıflandırıcıların bir kombinasyonunun kullanılması ve yeni veri hazırlama ve değerlendirme yöntemlerinin geliştirilmiş olmasıdır.

Toplamda, 6 ana, 55 altfamilya, 163 alt-altfamilya ve 634 alt katman EC sınıfı sınıflandırıcısından oluşan 858 EC sınıflandırıcısı eğitilmiştir. Doğrulama veri setlerini kullanarak tüm EC sınıfları için 0.99'luk ortalama F-ölçütü elde edilmiştir. Enzim veya enzim olmayan sınıflandırması, hiyerarşik bir tahmin yaklaşımı ile birlikte *ECPred*'e dahil edilmiştir. Bildiğimiz kadarıyla, Seviye 0'dan (enzim/enzim-olmayan) başlayıp 4. Seviyeye (alt katman sınıfı) kadar proteinlerin enzimatik fonksiyonunu tahmin eden ilk çalışma budur. Son olarak, *ECPred* bağımsız test setleri üzerinden diğer benzer araçlarla karşılaştırıldı ve *ECPred* mevcut araçlardan daha iyi sonuçlar elde etti, ancak sonuçlar iyileştirme için hala çalışma yapılabileceğini göstermektedir.

Anahtar Kelimeler: Enzim, Enzim Komisyonu Numarası, Makine Öğrenmesi, Sekans Analizi

To my family

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor Prof. Dr. Mehmet Volkan atalay for his guidance, support and patience throughout this thesis. I also would like to thank my co-supervisor Prof. Dr. Rengül Çeytin-Atalay, for her helpful advice, criticism about biological aspect of this work. I consider myself very lucky to be working with them.

I am deeply indebted to Dr. Tunca Doğan, for his valuable comments, advises and constructive critiques to improve my thesis. I also would like to sincerely thank to Ahmet Rifaioğlu for helping to understand concept of the problem and his valuable suggestions to solve the problems when I come to deadlock.

I would like to thank my friends, Samet Sezek, Fatih Calip, Anıl Çetinkaya, Çağrı Kaya, Alperen Eroğlu, Gökhan Özsarı, Alper Karamanlıoğlu, Tuğberk İşyapar, Arınç Elhan and Özcan Çataltaş. It's been always great to spend time with you.

Finally, I would like to thank my family for their endless support. I am forever indebted to my father Mustafa Dalkıran and my mother Yasemin Dalkıran. I also would like to thank my brothers, Ahmet and Ali Furkan, for their support during my master thesis. Finally, I would like to thank my grandfather, Enver Akça, for encouraging and motivating me during my study.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

DNA	Deoxyribonucleic acid
EC	Enzyme Commission
GO	Gene Ontology
SVM	Support Vector Machine
*k*NN	*k*-Nearest Neighbourhood
ANN	Artificial Neural Network
NB	Naive Bayes
RF	Random Forest
NA	Not Available
WT	Web-based Tool
DT	Desktop-based Tool
UniProtKB	Universal Protein Resource Knowledge Base
PDB	Protein Data Bank
KEGG	Kyoto Encyclopedia of Genes and Genomes
OMIM	Online Mendelian Inheritance in Man
DBGet	Integrated Database Retrieval System
PSSM	Position Specific Scoring Matrix
OET-*k*NN	Optimized Evidence-Theoretic *k*-Nearest Neighbourhood
AAC	Amino Acid Composition
Am-Pse-AAC	Amphiphilic Pseudo-Amino Acid Composition
RBF	Radial Basis Function
AFK-NN	Adaptive Fuzzy *k*-Nearest Neighbourhood
MOLMAP	MOLecular Mapping of Atom-level Properties
SOM	Self Organizing Maps
CTF	Conjoint Triad Feature
RBFSVM	Adaboost Algorithm with SVM with RBF Kernel
AM-SVM	Arithmetic Mean Offset SVM
MCC	Matthew’s Correlation Coefficient

BR-*k*NN	Binary Relevance *k*-Nearest Neighbourhood
MCC	Multiple Sequence Alignment
MCC+SS	Multiple Sequence Alignment Secondary Structure
ECOH	Enzyme COmmission Number Handler
MCS	Maximal Common Structure
MI	Mutual Information
MTTSI	Maximal Test to Training Sequence Identity
ACC	Autocross Covariance
AC	Autocross
CC	Cross Covariance
BLAST-*k*NN	BLAST *k*-Nearest Neighbourhood
EMBOSS	European Molecular Biology Open Software
SPMAP	Subsequence Profile Map
ROC	Receiver Operating Characteristic
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
PFAM	Protein Families Database

CHAPTER 1

INTRODUCTION

Proteins are large biomolecules that play essential roles in living cells and they consist of amino acids. Proteins perform a lot of functions such as catalyzing biochemical reactions, replication of DNA, intracellular transport and protecting the body from viruses and bacteria.

Ontological systems are defined by consortiums such as Gene Ontology (GO) and Enzyme Commission (EC) Nomenclature in order to provide a vocabulary to represent the relationships among entities. GO and EC are the special type of biological ontologies that annotate functions of proteins and enzymatic functions of proteins, respectively. Protein functions are basically determined by experiments such as analysis of microarrays and RNA interference. The Universal Protein Resource (UniProt) is a database which provides sequence and functional information of proteins. In UniProt, curators search the literature and gather the information related to a protein and introduce the information to the research community.

1.1 Problem Statement

Automated protein function prediction can be defined as a method that aims to assign automatically one or more functions to a given protein. While the number of protein sequences is increasing rapidly, manual annotation of functions to proteins cannot catch up with this number. It is necessary to develop systems to predict automatically protein functions since the manual annotation is both time-consuming and costly. Several methods have been proposed in the literature to predict automatically functions of protein. Most of the methods use the protein sequence or protein structure to detect functionality. Predicting enzymatic functions of proteins is one of the important topics in bioinformatics since enzymes play important roles in the metabolism by catalyzing biochemical reactions. Enzyme Commission (EC) numbers are ontology terms in the form of numerical representations, describing enzymatic functions based on chemical reactions that they catalyze. EC numbers consist of six main classes (i.e. oxidoreductases, transferases, hydrolases, lyases, isomerases and ligases) and their subclasses on four hierarchical levels in total [1]. One basic problem in this field is predicting whether a protein is an enzyme or not and this subject is overlooked in most of the studies. The information concerning a protein being an enzyme can then be used to predict specific enzymatic activities of proteins in a hierarchical manner. In this thesis, we pursue a machine learning approach and construct binary classifiers in order to tackle the problem. Positive and negative datasets are necessary in order to construct a binary classifier. Another basic problem in existing studies is about constructing negative datasets is to simply perform it by selecting proteins that are not in the positive dataset and this approach has several problems . Enzymatic functions are not popularly studied in the literature, however, the hierarchical structure of EC is quite suitable for automatic function prediction. Most of the studies are limited to predicting first two or three levels of the hierarchy. This topic was previously studied by our group members [2] [3], however there weren't any independent test set and enzyme/non-enzyme discrimination wasn't applied.

1.2 Approach

In this thesis, we present a novel method called *ECPred* to predict firstly, whether a protein sequence is an enzyme or a non-enzyme by constructing six classifiers, each corresponding to one of the six main EC classes, with a combinatorial machine learning approach. The idea is that if all six classifiers give low prediction scores for a given input protein sequence, it can be labeled as non-enzyme, whereas if the target protein receives a prediction score higher than the class specific cut-off value, it is predicted to be an enzyme with the corresponding basic enzymatic function. After deciding main EC class of protein, its subfamily, sub-subfamily and substrate classes are predicted subsequently. We constructed positive and negative training datasets using proteins which are annotated with an EC number and proteins that have not been annotated with an EC number in UniProtKB/Swiss-Prot database, respectively.

ECPred combines three independent classifiers: SPMap, BLAST-*k*NN and Pepstats-SVM that are based on subsequences, sequence similarities, and amino acid features, respectively; similar to the method developed previously by our research group for protein function prediction: GOPred [4]. For the training of the SPMap classifier, fixed-length subsequences are extracted from protein sequences in the positive training data and the subsequences are clustered based on their similarities. Feature vectors are then generated using profiles of subsequences. Proteins that are converted into feature vectors are given as an input to Support Vector Machine (SVM) classifier. BLAST-*k*NN is used to get *k*-nearest sequences from positive and negative training datasets based on pairwise BLAST scores and a similarity score is calculated for each input sequence. Pepstats-SVM converts protein sequences into 37-dimensional feature vectors by extracting their physicochemical peptide statistics. These converted sequences are subsequently fed to the SVM classifier as an input. The proposed system combines these three methods and it gives a weighted mean score for each EC class.

Proteins available in UniProtKB/SwissProt database are used as the training data. EC numbers which had more than 50 protein associations are chosen for training by*ECPred*. Totally, 858 EC class classifiers are trained: six main EC class classifiers, 55 subfamily classifiers, 163 sub-sub classifiers and 634 substrate classifiers.

1.3 Improvements

Major improvements brought by this thesis are as follows:

- Enzyme or non-enzyme classification is incorporated into *ECPred* along with a hierarchical prediction approach. To the best of our knowledge, this is the first study that predicts the enzymatic function of proteins starting from Level 0 (enzyme/non-enzyme) going down to Level 4 (substrate class).

- *ECPred* has achieved an average F-score value of 0.99 on validation datasets.

In addition to the above mentioned major improvements, we describe a method to construct positive and negative datasets such that they are balanced and their sizes are reasonable for training. The number of trained EC class classifiers is 858. We provide positive and negative cut-off values which are determined separately for each EC class. In this study, the size of the independent test dataset is not huge, however, comparisons are extensively made with available web-based tools.

CHAPTER 2

BACKGROUND INFORMATION AND RELATED WORK

In this section, background information about enzyme is given. Totally, 20 existing studies are examined.

2.1 Enzymes

Proteins are biomolecules that play important roles in the body. Proteins perform functions such as: catalyzing biochemical reactions, DNA replication, transporting molecules from one location to another within the cell. Enzymes are one type of proteins which speed up biochemical reactions by lowering activation energy. Enzymes have an active site, where the biochemical reaction happens. Substrates are specific kinds of molecules that are bound to active sites of enzymes to initiate biochemical reactions. When enzymes bind substrates, an enzyme-substrate complex is formed. Finally, enzyme-substrate complex breaks into enzyme and products. Enzymes can take place in more than one biochemical reactions because the structure of an enzyme doesn't alter after reaction. An illustration of an enzymatic reaction is given in Figure 2.1.

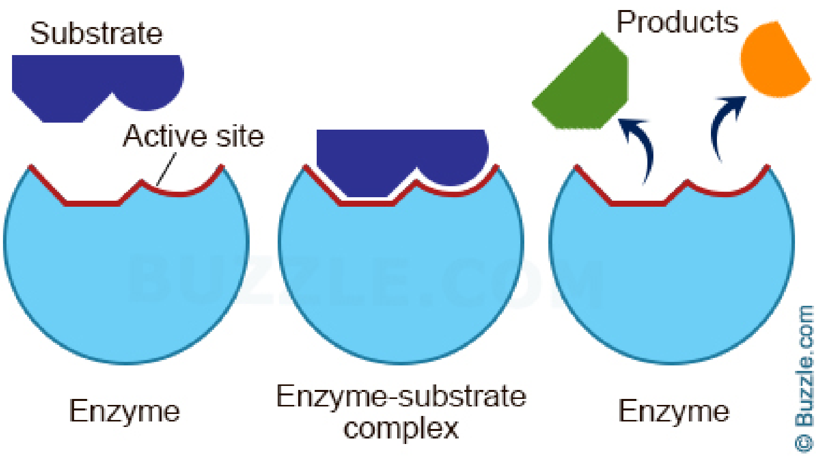


Figure 2.1: An illustration of an enzyme-substrate relation. In the first step, a substrate enters the active site of the enzyme. In the second step, an enzyme-substrate complex is formed. In the final step, two products are created. Adopted from “Enzymes Cont’d.” Biochem80p. biochem80p, Trinidad and Tobago, 12 March 2014. Web. 18 July 2017.

2.2 Structure of Enzyme Commission Nomenclature

Nomenclature Committee of the International Union of Biochemistry classifies enzymes according to the reaction they catalyze. Enzyme Commission (EC) numbers are the numerical representation of enzymes based on this classification. EC numbers are represented as four elements separated by periods. The first digit indicates which of the six EC classes it belongs, the second digit represents the subfamily class, the third digit expresses the sub-subfamily class and the fourth digit shows the substrate information of the enzyme in its sub-subclass [1]. For an EC Number, EC 4.2.3.1, EC 4 represents EC class 4 (Lyases), EC 4.2 is carbon-oxygen lyases, EC 4.2.3 is carbon-oxygen lyases that act on phosphates and EC 4.2.3.1 is carbon-oxygen lyases that act on phosphates where threonine synthase is one of the substrates of this enzyme. The hierarchical tree structure of EC numbers is presented in Figure 2.2. EC numbers are separated into six main classes according to the biochemical reactions they catalyze. An EC number should carry functions of its parents because there is an *is-a* relationship between the EC numbers. Some enzymes contain more than one catalytic activities and annotated with more than one EC numbers. These enzymes called multi-functional enzymes.

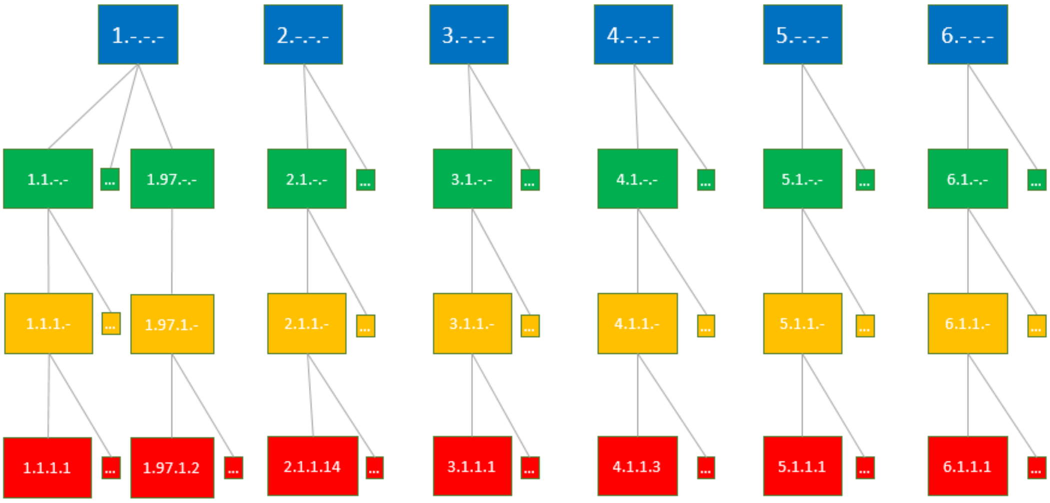
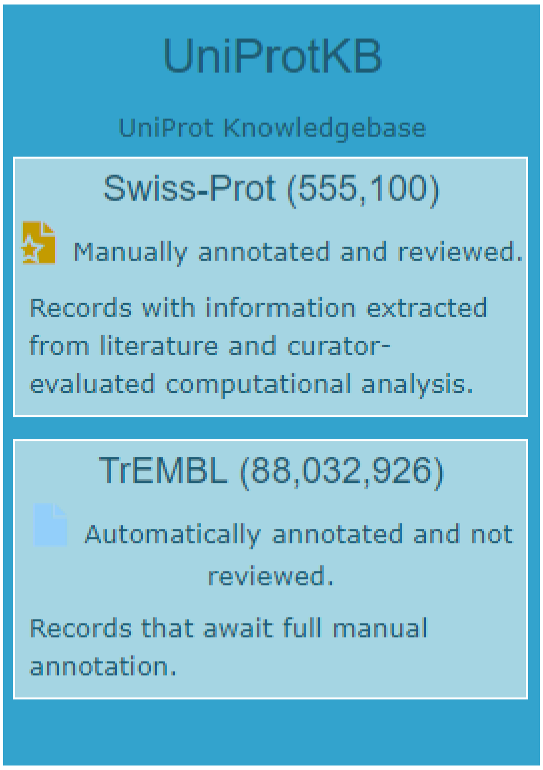


Figure 2.2: Hierarchical tree structure representation of EC numbers.

2.3 Universal Protein Resource Knowledge Base (UniProtKB)

UniProtKB is a protein database for comprehensive protein information such as function, enzyme specific information, subcellular location, classification, etc. It consists of two sections which are given in Figure 2.3. The first section is UniProtKB/Swiss-Prot which is reviewed and manually annotated while the second section is UniProtKB/TrEMBL which is automatically annotated and is not reviewed.



UniProtKB

UniProt Knowledgebase

Swiss-Prot (555,100)

Manually annotated and reviewed.

Records with information extracted from literature and curator-evaluated computational analysis.

TrEMBL (88,032,926)

Automatically annotated and not reviewed.

Records that await full manual annotation.

Figure 2.3: UniProtKB database consists of two parts. Swiss-Prot contains 555,100 proteins and TrEMBL contains 88,032,926 proteins. The screenshot is taken from "UniProt" UniProt. UniProt, EMBL-EBI, 5 July 2017. Web. 18 July 2017.

2.4 Literature Survey on Enzyme Classification

There exist several studies on classifying enzyme functions based on the EC hierarchy level. In this study, we denote the levels of enzyme function classifications as follows:

- Level 0: enzyme or non-enzyme;
- Level 1: enzyme main family;
- Level 2: enzyme subfamily class;
- Level 3: enzyme sub-subfamily class;
- Level 4: enzyme substrate family class.

In this section, we present a non-comprehensive survey of the existing studies. All of the studies were analyzed according to

- the computational method employed for classification,
- the level of enzyme function classifications and
- the input feature and dataset size.

As seen from Table 2.1 Support Vector Machines (SVM) and *k*-Nearest Neighbor (*k*NN) are the most popular computational methods that are employed for enzyme classification. There are also a few studies that use Artificial Neural Networks (ANN) and Random Forests (RF). In most of the studies, only enzyme or non-enzyme discrimination (Level 0) and identification of six main enzyme classes (Level 1) have been studied. However, there are some studies that classified subfamily classes (Level 2) or sub-subfamily classes (Level 3). We have come across only two studies that predicted the whole EC nomenclature. In general, the sequence information was obtained from ENZYME database (http://enzyme.expasy.org/) and UniProtKB/SwissProt database
(http://www.uniprot.org/uniprot). However, we observed that in some studies, PDB (https://www.rcsb.org/pdb) and KEGG Ligand database
(http://www.genome.jp/kegg/ligand.html) have also been used to construct training or test dataset.

Input feature extraction methods can be divided into four categories: homology-based approaches, subsequence-based approaches, feature-based approaches and structural based approaches. The assumption is that, since the homologous protein sequences are similar to each other, they would have the same functions. Homology-based approaches use this assumption to detect similar enzyme functionalities. A high-level sequence homology is usually considered to be a powerful sign of functional homology. Subsequence-based methods focus on important regions of sequences such as domains and motifs that are highly related to the functions of corresponding proteins. When the annotations to be associated needs a certain motif or domain, these methods become quite effective. In feature-based methods, biological features such as the number of residues, isoelectric point, charge and the further chemical features are calculated from the protein sequence. In general, structural similarity between two proteins indicates similar functions because protein structure is usually better conserved than the protein sequence. Therefore, structural based approach is one of the most popular approaches in protein function prediction. The above mentioned methods are summarized in Table 2.1 In the rest of this Section, "success rate" is used as a generic term to indicate the performance of a given system. However, the calculation of success rate may be different from one study to another.

Jensen *et al.* [5] proposed a system to detect and classify enzymes from their sequence. Unlike the traditional methods, which uses similarity of sequence, they used post-translational modifications and localization features such as subcellular location, secondary structure and low complexity regions. Chromosomal gene locations were taken from Online Mendelian Inheritance in Man (OMIM) database thorough SwissProt reference links, UniProtKB/SwissProt database was used the extract the training dataset. Totally, 5,658 protein sequences were firstly classified at Level 0 and then, annotated with one of the six main EC classes (Level 1). An artificial neural network (ANN) was used as the classifier. When sensitivity rate is below 40%, they obtained low false-positive rate based on cross validation. The number of samples was not sufficient and the authors discriminated input sequences at Level 0 and Level 1.

Dobson and Doig [6] proposed a system to discriminate Level 0 proteins without using alignment. Training dataset consisted of 1,178 proteins which split into 691 enzymes and 487 non-enzymes. All proteins were taken from Protein Data Bank (PDB) and represented by using 52 features such as secondary structure fractions, residue option, residue surface, existence of ligands and the size of the biggest surface pocket. SVMs were used to classify the proteins at Level 0. 77% accuracy rate was reported for enzyme or non-enzyme prediction. When the dimension of the feature vector was reduced to 36, the accuracy rate was increased to 80%. The authors extended their system to predict the Level 1 of a given protein based on the same method. In the extended study [7], called as Integrated Database Retrieval System (DBGet), ENZYME and Astral SCOP databases were employed to construct the training and test datasets. 498 protein sequences were obtained in total. One-versus-all SVMs were combined to obtain the predictions. According to the jackknife test results, 60% success rate was achieved with top two ranks (the correct main class was in the top two highest

Table 2.1: Summary of the methods mentioned in this section. Classifier Types; **SVM**: Support Vector Machines, **ANN**: Artificial Neural Networks, ***k*NN**: *k* Nearest Neighborhood, **NB**: Naive Bayes and **RF**: Random Forest. Level (Enzyme function classification Level); **0**: Enzyme or non-enzyme, **1**: Main Class, **2**: Subclass, **3**: Sub-subclass and **4**: Substrate. Input Feature Extraction Methods; **a**: Homology based, **b**: Feature based, **c**: Subsequence based and **d**: Structural based. Tool Availability; **NA**: Not available, **WT**: Web-based tool and **DT**: Desktop tool.

Reference	**Classifier**	**Level**	**Performance (%)**	**Input Feature**	**Dataset Size**	**Tool Avail.**
[5]	ANN	0-1	40	b	5,658	WT
[6]	SVM	0	77	d	1,178	NA
[7]	SVM	1	60	d	498	NA
[8]	NB	1	45	d	498	NA
[9]	*k*NN	0-2	92	c	19,682	WT
[10]	SVM	2	81	b	2,640	NA
[11]	*k*NN	2	92	b	252,625	NA
[12]	SVM	0-1	91; 95	c	7,329	NA
[13]	*k*NN	1	99	b	1,200	NA
[14]	SVM	0	97	b	2,400	NA
[15]	RF	1-3	92	b	3,741	NA
[16]	SVM	2	93	b	NA	NA
[17]	SVM, *k*NN	0	86	d	1,177	NA
[18]	SVM	2	98	b	NA	NA
[19]	*k*NN	1-4	98	b	300,747	DT
[20]	RF	1-3	98	b	7,131	NA
[21]	ANN	1	96	d	6,081	NA
[22]	SVM	3	99	d	5,643	DT
[23]	RF	1-4	98	a,c	1,121	NA
[24]	*k*NN	1	94	d	59,763	WT
ECPred	SVM,*k*NN	0-4	99	a,b,c	245,209	NA

scored predictions). These two studies were performed with a very low number of protein sequences and they are limited to Level 0 and Level 1 of the EC hierarchy. Furthermore, there is no available tool.

Borro *et al.* [8] proposed a system to predict Level 1 using Naive Bayes classifier. In order to compare the methods, they used the same set of protein structures which was employed by Dobson and Doig [7]. All of the structure information was taken from PDB database and 498 proteins were selected in total for training dataset. Their system consisted of three parts. Firstly, in order to obtain which features were the most powerful, they calculated the correlation matrix amongst all protein features. In the second step, they checked whether these features were also correlated in the complete database. Finally, redundant features were removed to decrease the noise in the data. After constructing features, they ran the Naive Bayes classifier using Weka [25]. According to the ten-fold cross validation results, 45.3% accuracy was achieved. This study was limited to predict only the Level 1 with a small dataset.

Shen and Chou [9] developed a web tool which predicted Level 1 and Level 2 of the EC hierarchy using a top-down approach. Functional domain information was used to construct Pseudo Position-Specific Scoring Matrix (Pse-PSSM). Each protein was represented as an 8,958-dimensional vector. ENZYME database was used to construct the dataset for enzyme main class (Level 1) and subfamily class (Level 2) while the functional domain information was taken from Pfam database. Totally, 19,682 protein sequences were obtained, which consisted of 9,832 enzyme sequences and 9,850 non-enzyme sequences. The Optimized Evidence-Theoretic *k*-nearest neighbor (OET-*k*NN) was used as the classifier which was previously applied to the subcellular localization problem. According to the jackknife results, the overall success rate was 91.3% on discrimination of Level 0 and the overall success rate for identifying Level 1 was 93.7%. Finally, the average success rate for subfamily classes of oxidoreductase, transferases, hydrolases, lyases, isomerases, and ligases were 86.7%, 95.8%, 95.9%, 94.4%, 93.3%, and 98.3%, respectively. They worked on Level 2 identification and their set size is not too small but also not big enough for testing. A web-based tool is available which gives a three level (level 0, level 1 and level 2) predictions for a given protein sequence.

Zhou, Chen, Li and Zou [10] developed a system to predict Level 2 using SVMs. As an input feature, they used Chou's amphiphilic pseudo-amino acid composition (Am-Pse-AAC) [26] features which were the modified version of AAC. The difference was that Am-Pse-AAC used hydrophobic and hydrophilic values of amino acids. The dataset was constructed from SWISSPROT database: 2,640 oxidoreductase sequences (Class 1) and 16 subfamily classes were obtained. Firstly, they compared different kernel functions for SVMs. According to the 5-fold cross validation results, linear kernel achieved 52.65% accuracy, polynomial kernel achieved 72.95% accuracy and finally, RBF kernel achieved 78.37% accuracy. The authors also compared their methods with the existing studies: CDA [26] and AFK-NN [11], which were also proposed based on Am-Pse-AAC. According to the jackknife test, the author's method obtained 80.87% which is 10% higher than CDA and 4% higher than AFK-NN. This study comprised only oxidoreductase (Class 1) and the dataset is small for testing.

Huang, Chen, Hwang and Ho [11] proposed a study to predict Level 2 of the EC hierarchy using an adaptive fuzzy *k*-nearest neighbor (AFK-NN) classifier. 252,625 proteins were selected from ENZYME database and UniProtKB/SwissProt for training dataset. As the input features, the authors used amphiphilic pseudo-amino acid composition (Am-Pse-AAC) which was the modified version of amino acid composition (AAC). In this version, hydrophobic and hydrophilic amounts were added to AAC as new components. C5.0 decision tree algorithm and SVM were used to make comparisons with the proposed method AFK-NN. Overall accuracy of 92.1% was achieved according to the jackknife test which was slightly better than C5.0 (91.2%) and SVM (91.7%) alone. The authors also compared their method with previous studies of Chou and Elrod [27] and Chou [26] on the same dataset. According to the jackknife test, Chou achieved 70.61% accuracy using CDA as the input feature, AFK-NN achieved a better result with 74.88% accuracy. Although the dataset size was sufficient, only Level 2 predictions were performed in this study.

Lu, Qian, Cai and Li [12] developed a web-based system which predicts first Level 0 of the EC hierarchy. The system then predicted which of the six EC main classes (Level 1) it belonged to if it was an enzyme. For each input protein sequence, a 2,657-dimensional feature vector was generated using the protein's functional domain information from Pfam database. The feature vectors were then input to a support vector machine (SVM) classifier. The positive training dataset was constructed using ENZYME database while the negative training dataset was generated based on the UniProtKB/SwissProt database. 2,443 proteins were obtained among 70,573 proteins after applying some filters in order to construct positive training dataset. 4,886 random proteins were selected among 145,271 proteins for the negative training dataset. According to the jackknife test, the authors classified proteins as enzymes or non-enzymes with 86% success rate and the overall success rate was 91.32% for six main EC classes. They developed a web-based tool, however, it is currently not available. The drawbacks of this study are that the number of proteins for training (2,443) is low and the predictions are given for only the first level of the EC hierarchy.

Nasibov and Kandemir-Cavas [13] made an efficiency analysis of *k*NN and the minimum distance-based classifiers on the Level 1 prediction. 200 proteins were selected for each class. In this study, the authors used training and test dataset with different percentages and they achieved the maximum accuracy when 25% of the proteins are kept as the test dataset. All protein sequences were taken from ENZYME database. A protein sequence was encoded as 1 by 20 vector where each element of the vector represented the frequency of amino acids of the protein sequence. Two modified versions of *k*NN were proposed. In the first one (method 1), the distance of the test enzyme from the average frequency of amino acid of each class was computed and the test enzyme class was assigned to the nearest one. In the second method, the same distance was calculated by adding the amino acid frequency of test class. They computed distance score between these added frequencies and previously calculated frequencies (method 1) and the test enzyme was labeled with the class with minimum distance score. According to the performance results, both approaches achieved overall accuracy of 95% and *k*NN with *k*=6 achieved 99% of accuracy. Since there is no ideal solution to find the value of *k* and it is calculated experimentally and by the error rate, the execution time of *k*NN algorithm was much longer than the two proposed methods. Although the dataset size was sufficient, only Level 1 predictions were performed in this study.

Qiu, Luo, Huang and Liang [14] developed a system that used the discrete wavelet transform based on the chemical features of residues as the features and SVMs to classify the proteins at Level 0. The authors employed the same dataset of 1,178 proteins that Dobson and Doig [6] used which consisted of 691 enzymes and 487 non-enzymes. In addition, they made use of a second dataset for testing which consisted of 1,200 enzymes and 1,200 non-enzymes where all of the proteins have sequence similarity less than 40%. 96.96% and 97.74% accuracy rates were achieved for enzyme and non-enzyme predictions, respectively. The dataset size was small compared to the previous studies and only Level 0 prediction was performed.

Latino and Aires-de-Sousa [15] proposed a system to predict Level 3 using MOLecular Mapping of Atom-level Properties (MOLMAP) reaction descriptors applying RF. MOLMAP reaction was obtained by the change between the product's MOLMAP and reactant's MOLMAP. All the enzymatic reactions were taken from KEGG LIGAND database. Initially, they started with 6,810 reactions and after the elimination process, they obtained 3,741 reactions (7,482 when represented in both ways). Self Organizing Maps (SOM) were used to generate a molecular descriptor. After the calculation of MOLMAP descriptors, RF was used classify Level 1, Level 2 and Level 3. According to the independent test dataset results, they correctly assigned 95%, 90% and 85% of enzyme main family class (Level 1), enzyme subfamily class (Level 2) and enzyme sub-subfamily class (Level 3), respectively. This study was performed with a low number of reactions and it was limited to classification of Level 3 of the EC hierarchy. Moreover, there is no available tool.

Wang, Wang, Yang and Deng [16] proposed a system to predict Level 2 using two modified versions of SVMs. The authors used Conjoint Triad Feature (CTF) to construct input features which were the modified version of amino acid composition (AAC). In CTF, 20 amino acids were divided into seven different classes based on their dipoles and amount of the side chains. Each protein was represented as a 343-dimensional vector (7*7*7) where each member of this vector was the density of the CTF occurrence in the enzyme sequence. Totally, 43 enzyme subfamily classes were trained for this study. Two adapted versions of SVMs; AdaBoost algorithm with SVM with RBF kernel (RBFSVM) and SVM with arithmetic mean offset (AM-SVM) were compared to investigate the performance of their studies. According to the ten-fold cross validation result, AM-SVM achieved 92% for Matthew's correlation coefficient (MCC) and AdaboostSVM obtained 83% for MCC. They also compared features

AAC and CTF using AM-SVM methods on oxidoreductases' subfamily classes and the results showed that except two subfamily classes AM-SVM with CTF obtained a better result. There is no information about the dataset size in this study.

Davidson and Wang [17] developed a novel ensemble method to predict Level 1 which consisted of three SVMs and two *k*NN algorithms where they used the majority voting rule. A dataset of 697 enzymes and 480 non-enzymes was constructed from the study of Dobson and Doig [6]. The authors employed the same 52 features of Dobson and Doig which consisted of five main parts: residues percentage, surface area percentage, heterogeneous number, secondary structure percentage and others. They also used four more features: magnesium ions count, the total number of residues, surface area and surface pocket counts. A success rate of 85% was achieved in a ten-fold cross validation and 86% success rate in jackknife test. No tool is available for this study. The number of proteins in this study's dataset was low and this study was limited to the classification of Level 1 of the EC hierarchy.

Wang et al. [18] proposed another system, this time they predicted Level 3 using a modified version of SVM which they extended from their previous study. CTF was used again as the input feature. A dataset of proteins with sequence identity less than 40% were constructed. EC sub-subfamily classes which contained at least 50 proteins were included in the training dataset. Six main classes and eighty five sub-subfamily class were trained. The authors proposed a modified version of the PMSVMHL (which is a different version of the Hierarchical Max-Margin Markov [28] by employing zero-one loss) method called SVMHL which consumed less time than PMSVMHL. SVMHL, PMSVMHL and the standard SVM were compared on a simple dataset. PMSVMHL and SVMHL achieved better results than the standard SVM and the training time was reduced 16 times in comparison to the PMSVMHL method. The authors also compared their previous methods AM-SVM and SVMHL on EC sub-family dataset. SVMHL outperformed AM-SVM method except for one sub-class. According to the 10-fold cross validation results, 91% MCC and 98% accuracies were obtained in predicting six main classes. 92% and 82% MCC values were obtained in predicting subclasses and sub-subclasses, respectively. As in their previous studies, there was no available tool and no information about dataset size, but this time the authors worked on Level 3 of the EC hierarchy.

Ferrari, Aitken, Jano and Goryanin [19] developed a system called EnzML which predicted multi-functional Level 4 of the EC hierarchy using InterPro signatures. The protein sequences and their EC annotation was taken from UniProtKB/Swiss-Prot, UniProtKB/TrEMBL, InterPro, KEGG and ExPASy ENZYME database. Each protein sequence was represented as the presence and absence of InterPro signatures. They selected sequences from SwissProt and KEGG having the same annotation on both databases. The final set contained 300,747 proteins, 55% were enzyme sequences and 45% were non-enzyme sequences. They used Binary Relevance *k*-Nearest Neighbor (BR-*k*NN) as classifier. According to the cross evaluation results, they obtained 98% accuracy for the exact match of the all 4 levels.

Kumar and Choudhary [20] proposed a system to predict up to Level 2 of the EC hierarchy using a Random Forest (RF) algorithm. In order to construct input features, they used online tools EMBOSS-PEPSTAT [29] which computed 61 feature values and ProtParams [30] that generated 36 feature values. These features were combined and 73 input features were generated in total. 2,400 non-enzymes and 4,731 enzymes were taken from SWISS-PROT database to construct the training dataset. Proposed system consisted of two models. The primary model first predicted whether a sequence was an enzyme or not (Level 0); if it was an enzyme, the system classified the main EC class (Level 1). Finally, the system predicted the sub-family class (Level 2). According to the ten-fold cross validation results, overall accuracies of 94.87%, 87.7% and 84.25% were achieved for the Level 0 classification, Level 1 classification and Level 2 classification, respectively. In the second model, Level 2 of the EC hierarchy was directly predicted using the RF algorithm and an overall accuracy of around 87% was achieved. Finally, the authors ran an R package called Rattle to look for the importance of the input features. Cysteine percentage and molecular weight were found to be the top two most important attributes. This study was limited to predict only the first three levels of the EC hierarchy and dataset size was smaller than some of the previous studies.

Volpato, Adelfio and Pollastri [21] proposed a system to predict Level 1 of the EC hierarchy using artificial neural networks (ANN). Each protein sequence was represented by the residue frequency which was obtained from multiple sequence alignments. They only selected animal taxonomy group for this study and the dataset was constructed from ENZYME database which consisted of 6,081 protein sequences. PSI-BLAST was run three times in order to determine amino acid-residue frequency. The authors constructed two different datasets and they called these datasets Multiple Sequence Alignment (MSA) and MSA+SS (Secondary Structure), respectively. MSA+SS contained three additional input to the MSA dataset. The system was trained by ten-fold cross validation using an n-to-1 neural network. According to ten-fold cross validation results, they obtained MCC values 84% for MSA-dataset and 83% for MSA+SS-dataset. This study was limited to classify only the level 1 of the EC hierarchy with a small dataset.

Matsuta, Ito and Tohsato [22] developed a system to predict Level 3 using SVMs called Enzyme COmmision number Handler (ECOH). The proposed system consisted of three steps: in the first step, they extracted substructures from the substrates and products using maximal common structure (MCS) algorithm. In the second step, they calculated mutual information (MI) values from these extracted substructures. In the final step, they predicted EC number of target reaction using SVM. They used KEGG database to construct training dataset. Totally 5,643 reactions were obtained after elimination and these reactions covered 162 EC sub-subfamily classes (Level 3). According to the jackknife test results, they achieved 86.1% sensitivity, 87.4% precision and 99.8% accuracy. They also predicted multi-functional enzymatic reactions and 62.3% of reactions were correctly predicted. They developed a standalone tool, but it works on only Windows 32-bit device. Their reaction set is small and they worked on identifying Level 3 of the EC hierarchy.

Nagao, Nagano and Mizuguchi [23] developed a system for the first time for predicting Level 4 of the EC hierarchy applying the RF algorithm. Sequence similarities and residue similarities for active sites, ligand binding sites and conserved sites were used as input features. Protein sequences were taken from UniProtKB/Swiss-Prot database and their information about CATH domain region was taken from Gene3D database. Totally 1,121 enzymes and corresponding 306 CATH super-families were used in the dataset. They calculated the maximal test to training sequence identity (MTTSI) for each query and 8 different MTTSI range was evaluated for benchmarking their system. 80% of the dataset was randomly selected as the training dataset and remaining 20% of the dataset was used as the test set. According to the benchmark results, 0.98

precision, 0.89 recall and 0.93 F-score values were achieved. The dataset used in this study is small and there is no available tool.

Che, Ju, Xuan, Long and Xing [24] developed a web-based system to predict Level 1 of the EC hierarchy using *k*NN algorithm. Totally, 59,763 protein sequences were selected from UniProtKB/Swiss-Prot database to construct the training dataset. They used autocross covariance (ACC) as an input feature. Firstly, they constructed Position Specific Scoring Matrix (PSSM) matrix implementing PSI-BLAST for each protein sequence, where each row of the matrix showed the corresponding residue type of amino acid letter. Then, they transformed PSSM matrices into fixed-length vectors by calculating the correlation between any two features. ACC resulted in the combination of two variables. Correlation of the same feature between two residues measured by autocross (AC) variable where cross covariance (CC) variable measured the correlation of two different features between two residues. According to the five-fold cross validation results, they achieved 94.1% overall accuracy on predicting six main EC classes. They also performed multi-functional enzyme class prediction on a small dataset which consisted of 1085 proteins. This time, they obtained 91.25% accuracy. They developed a web-based tool and their training dataset size is sufficient. However, they performed only Level 1 prediction.

Several methods and tools were proposed to classify EC hierarchy levels. When we investigated the studies, we see that most of the studies were limited to classifying first three level of the hierarchy (Level 0, Level 1 and Level 2). Only two of the studies predicted Level 4 of the EC hierarchy. There is no available method that uses a top-down approach to classify enzymes started from Level 0 to Level 4. All of the studies were limited to use single input feature type, except one study. Most of the studies performed on very small datasets. Finally, in most of the studies, there is no available tool.

Yaman [2] who is one of the member of our research group, proposed a system to predict Level 1-3 of EC hierarchy using SPMap [31]. However, this study was limited to Level 3 of EC hierarchy, only SPMap was used as a predictor and there wasn't any independent test set. Rifaioglu [3] who is also one of the member of our research group developed a system to predict first 4 levels of EC hierarchy. He obtained average F-score value 0.96, however, enzyme/non-enzyme classification wasn't applied and there were no independent test set.

CHAPTER 3

DATASETS AND METHODS

3.1 Datasets in General

In this study, positive and negative datasets are divided into two: training dataset and validation dataset. 90% of the initial dataset is used for training. The remaining 10% is employed for validation. The validation dataset is used to measure the performance of the system and to determine the cut-off values for SVM parameters. Protein sequences and their EC Number annotations are taken from UniProtKB/Swiss-Prot Release 2017_3. UniProtKB/Swiss-Prot is used for establishing the training and validation dataset since it is manually annotated and more reliable than UniProtKB/TrEMBL. UniRef [32] clustering module is also used which clusters proteins from the UniProtKB based on their sequence similarities. UniRef consists of three modules: UniRef100, UniRef90 and UniRef50. All identical sequences and fragment sequences from any living cell are combined into a single UniRef record in UniRef100. UniRef90 and UniRef50 are constructed by clustering UniRef100 records at sequence similarity 90% and 50%, respectively using CD-HIT algorithm [33]. Each UniRef90 cluster has one entry that represents sequences from UniRef100. Similarly, each UniRef50 cluster has one record that represents sequences from UniRef90. UniRef50 cluster is used in order to balance the positive and negative training dataset sizes, since the negative dataset size is initially bigger than the positive dataset size. Constructing positive and negative dataset is one of the most important steps in classification problems. Firstly, all proteins that are associated with any of the EC classes are downloaded from UniProtKB/Swiss-Prot database. Subsequently, proteins that include fragment sequences and proteins that are associated with more than one EC

Table 3.1: Total number of subfamily classes, sub-subfamily classes, substrate classes and the number of proteins are given for each class.

Level 1	Total number of Level 2 classes	Total number of Level 3 classes	Total number of Level 4 classes	Total number of proteins
Oxidoreductases	20	56	96	32,203
Transferases	9	31	230	77,042
Hydrolases	9	39	149	52,496
Lyases	6	14	64	19,707
Isomerases	6	14	38	12,174
Ligases	5	9	57	26,254
Total	55	163	634	219,876

class are eliminated since these multi-functional enzymes may be confusing for training and we are not aiming to predict more than one class for a given protein. Then, all annotations are propagated to the parents of the annotated EC class, since there is an *is-a* relationship between EC classes. For example, if a protein is associated with EC number 1.2.3.4, then, that protein is also associated with EC number 1.2.3.-, EC number 1.2.-.- and EC number 1.-.-.-, respectively. Finally, EC classes that are associated with at least 50 proteins are selected for training dataset. 10% of the class dataset separated as a validation set and these proteins are never used in training process. Totally, 858 EC classes (including six main EC classes) are obtained. Table 3.1 shows the detailed information and the number of proteins for each main enzyme classes that are used in training dataset. More explanation about constructing positive and negative training datasets are given in Section 3.1.1 and 3.1.2.

Totally, 6028 EC classes are available at ENZYME database (http://enzyme.expasy.org/). Number of trained and number of existing class information at each EC Level is given in Table 3.2. The coverage at Level 3 and Level 4 is low, since most of the EC classes at those levels are associated with less than 50 proteins.

3.1.1 Positive Training Dataset Construction for EC Numbers

Constructing positive training datasets is relatively easy compared to constructing negative datasets. For each EC class, proteins that are associated with that EC class are added to the positive training dataset. Since Transferases and Hydrolases contain

Table 3.2: Total number of trained and existing EC classes and coverage of ECPred.

EC Level	Number of trained class	Number of existing class	Coverage (%)
Main	6	6	100
Subfamily	55	69	80
Sub-subfamily	163	297	55
Substrate	634	5656	11

Table 3.3: The number of protein sequences after the application of UniRef50 for Level 1 and non-enzymes.

Classes	Total number of proteins	Total number of proteins after UniRef50
Oxidoreductases	36,577	8,596
Transferases	86,163	20,398
Hydrolases	59,551	16,550
Lyases	22,368	3,570
Isomerases	13,615	2,878
Ligases	29,233	4,466
Non-enzymes	292,589	100,459

significantly more proteins than the other classes, the positive training dataset sizes of these two classes are decreased using UniRef50. The number of sequences after applying UniRef50 for each main enzyme classes are shown in Table 3.3. For each main EC classes (except Transferases and Hydrolases, since they contain relatively more proteins than other four EC classes), 10% of the UniRef50 proteins are removed from all dataset as a validation set and remaining proteins are used in training datasets. For Transferases and Hydrolases, all UniRef50 proteins are selected for positive training dataset after removing 10% of them as a validation set. Then, randomly chosen proteins are added to these selected positive training dataset proteins to round the training dataset size to 36,000. Dataset sizes of six main EC classes in each elimination step are given in Table 3.4. For the rest of 852 EC numbers, proteins that are associated with that EC numbers are added to the positive training dataset.

Table 3.4: Training dataset sizes of Level 1 classes before and after elimination of multi-functional proteins and removing test set. (*For Transferases and Hydrolases more detailed explanations are given above).

	Total number of proteins			
Level 1	**Before elim. of multi-functional proteins**	**After elim. of multi-functional proteins**	**Test set (10% of UniRef50)**	**Training Dataset Size**
Oxidoreductases	40,883	36,577	860	35,717
Transferases	98,686	86,163	2,091	36,000*
Hydrolases	69,727	59,551	1,655	36,000*
Lyases	25,377	22,368	357	22,011
Isomerases	14,659	13,615	288	13,327
Ligases	29,961	29,233	447	28,786

3.1.2 Negative Training Dataset Construction for Level 1

Theoretically, if the protein is not annotated with a specific EC Number, that protein can be included in the negative set for that EC class. Therefore, the negative set size becomes very unbalanced compared to the positive dataset size, since negative sets include more proteins than positive sets. In order to balance the sizes, negative data set sizes are reduced using UniRef50 results. In UniProtKB, each entry has an annotation score between 1 and 5. Annotation score of 5 means, the entry is well studied and associated with best-annotated proteins while annotation score of 1 means that entry with a basic annotation and not well studied. There are no proteins that we can say that protein is 100% non-enzyme. In UniProtKB/Swiss-Prot, there are proteins that have EC number annotations and proteins that have not been annotated with an EC number yet. We assume that the proteins that have not been annotated with an EC number can be treated as non-enzyme. Since we are not 100% sure that all of these proteins are actually non-enzyme, only the proteins that have annotation score of 4 or 5 is used to include in the negative training dataset. For each annotation score, the number of proteins are given in Table 3.5. 10% of these non-enzyme proteins is also set aside for the validation set.

For each class, the proteins in the other five classes and non-enzyme proteins are selected to construct negative training dataset. The same number of proteins in the

Table 3.5: Number of proteins for each annotation score.

Annotation Score (out of 5)	Number of non-enzyme Proteins
1	20,407
2	37,302
3	17,388
4	8,876
5	16,457

Table 3.6: Training dataset sizes of Level 1 classes before and after elimination of multi-functional proteins and removing test set. (*For Transferases and Hydrolases more detailed explanations are given above).

Enzyme Class	1.-.-.-	2.-.-.-	3.-.-.-	4.-.-.-	5.-.-.-	6.-.-.-	Non Enzymes	Total
1.-.-.-	-	2,600	2,600	2,600	2,600	2,600	23,000	36,000
2.-.-.-	2,600	-	2,600	2,600	2,600	2,600	23,000	36,000
3.-.-.-	2,600	2,600	-	2,600	2,600	2,600	23,000	36,000
4.-.-.-	2,600	2,600	2,600	-	2,600	2,600	9,000	22,000
5.-.-.-	1,500	1,500	1,500	1,500	-	1,500	6,000	13,500
6.-.-.-	2,600	2,600	2,600	2,600	2,600	-	16,000	29,000

positive dataset are selected for the negative dataset in order to make the training dataset balanced. The positive and the negative training dataset construction is shown in Figure 3.1. Classes, subfamily classes, sub-subfamily classes and substrates that are colored with green are included in the positive training dataset, other five classes and non-enzymes are colored with red for the negative training set. For each Level 1 class, total negative dataset size and how many samples are taken from the other five classes and non-enzymes are given in Table 3.6. Non-enzymes are primarily selected from the proteins that have annotation score of 5 and remaining non-enzymes are selected from he proteins that have annotation score of 4, if necessary. Main EC classes 1.-.-..-, 2.-.-..-, 3.-.-..-, 4.-.-..-, 5.-.-..-, 6.-.-..- stands for Oxidoreductases, Transferases, Hydrolases, Lyases, Isomerases, and Ligases.

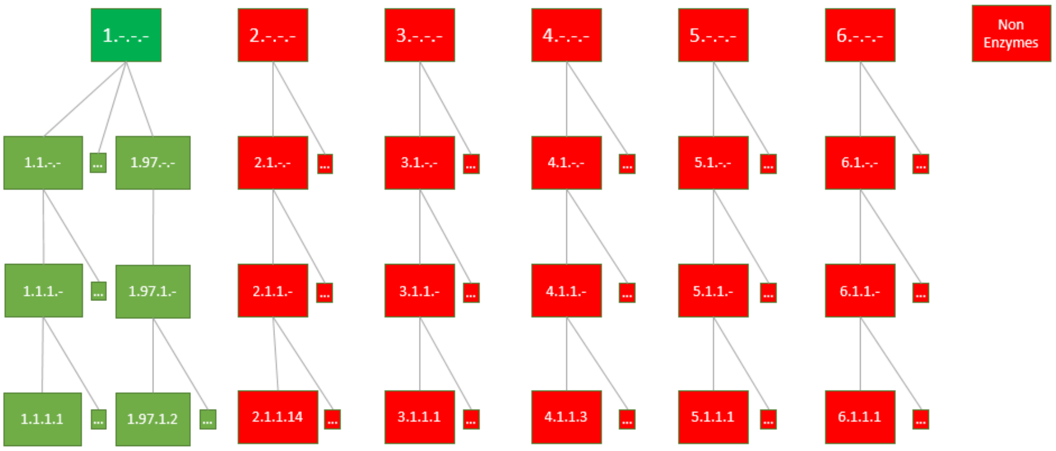


Figure 3.1: Positive and negative training dataset construction for EC class 1.-.-.-. Green color means that that class is used in the positive training set and red color means that that class is used in the negative training dataset.

3.1.3 Negative Training Dataset Construction for Level 2, Level 3 and Level 4

Certain rules are applied for constructing negative training datasets for Level 2, Level 3 and Level 4. Positive and negative training dataset constructions for Level 2, Level 3 and Level 4 are illustrated in Figure 3.2, Figure 3.3 and Figure 3.4, respectively. Green color means that that class is used in the positive training set, gray color means that that class is used neither in the positive training dataset nor in the negative training dataset and red color means that that class is used in the negative training dataset. The rules are as follows.

- For each class that has negative training dataset size greater than 10,000, half of its elements are taken from its siblings and their descendants, a quarter of its elements are selected from other five classes and a quarter of its elements are taken from non-enzymes for negative training dataset.
- For each class that has positive training dataset size between 1,000 and 10,000, same number of proteins with positive training dataset size from its siblings and their descendants, same number of proteins with positive training dataset size from other five classes (equally) and same number of proteins with positive

training dataset size from non-enzyme proteins are selected.

- For each class that has positive training dataset size less than 1,000, three times the number of proteins in the positive training dataset are selected from its siblings and their descendants, three times the number of proteins in the positive training dataset size are selected from other five classes (equally) and three times the same number of proteins in the positive training dataset size are taken from non-enzyme proteins.

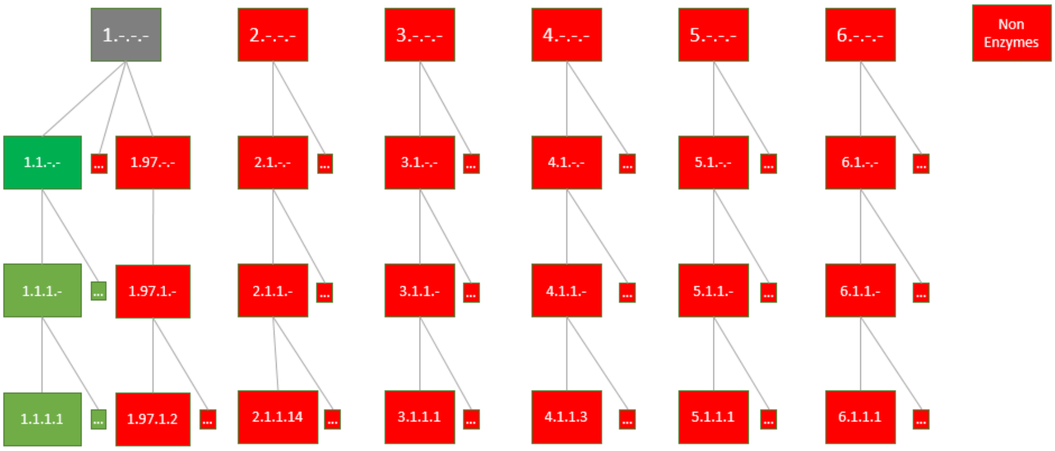


Figure 3.2: Positive and negative training dataset construction for EC class 1.1.-.-. Green color means that that class is used in the positive training set, gray color means that that class is used neither in the positive training dataset nor in the negative training dataset and red color means that that class is used in the negative training dataset.

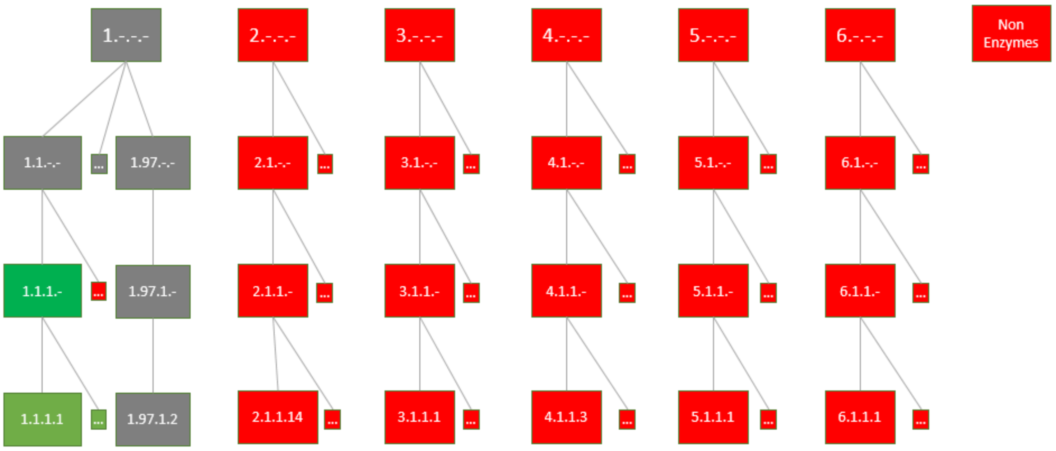


Figure 3.3: Positive and negative training dataset construction for EC class 1.1.1.-. Green color means that that class is used in the positive training set, gray color means that that class is used neither in the positive training dataset nor in the negative training dataset and red color means that that class is used in the negative training dataset.

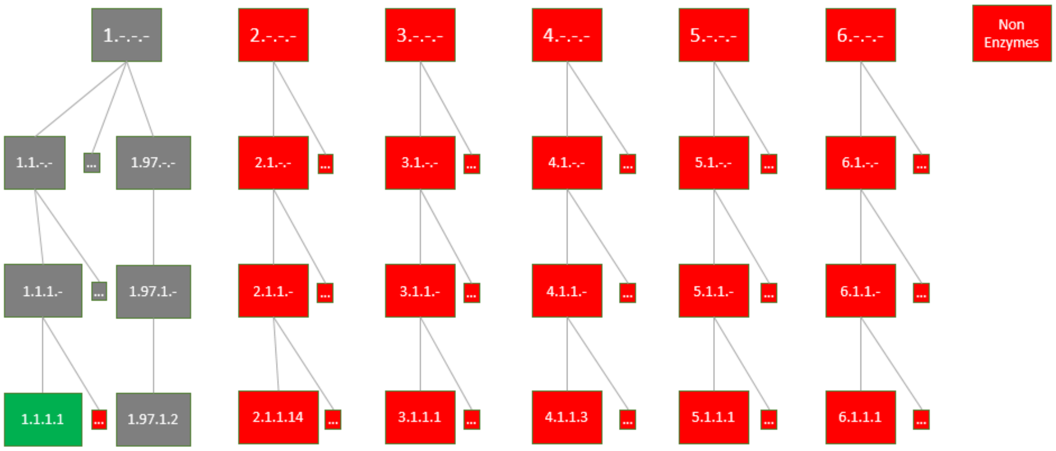


Figure 3.4: Positive and negative training dataset construction for EC class 1.1.1.1. Green color means that that class is used in the positive training set, gray color means that that class is used neither in the positive training dataset nor in the negative training dataset and red color means that that class is used in the negative training dataset.

3.2 Methods

GOPred [4] has been previously developed and it consists of three methods; the first method is BLAST *k*-nearest neighbor (BLAST-*k*NN) which is based on homology and BLAST score of *k*-nearest neighbors is used for prediction. The second method is PEPSTATS-SVM which is a feature based methods where peptide statistics are used. The third method is Subsequence Profile Map (SPMap) which is based on subsequence and proteins are classified based on their subsequences. All three methods are re-implemented in Java for this study.

3.2.1 BLAST-*k*NN

In order to classify a target protein, the *k*-nearest neighbor algorithm is used. Similarities between the target protein and proteins in the training dataset are calculated using the NCBI-BLAST tool [34]. *k*-nearest neighbors with the highest *k* BLAST score are extracted. The output of BLAST-*k*NN, O_B for a target protein, is calculated as follows:

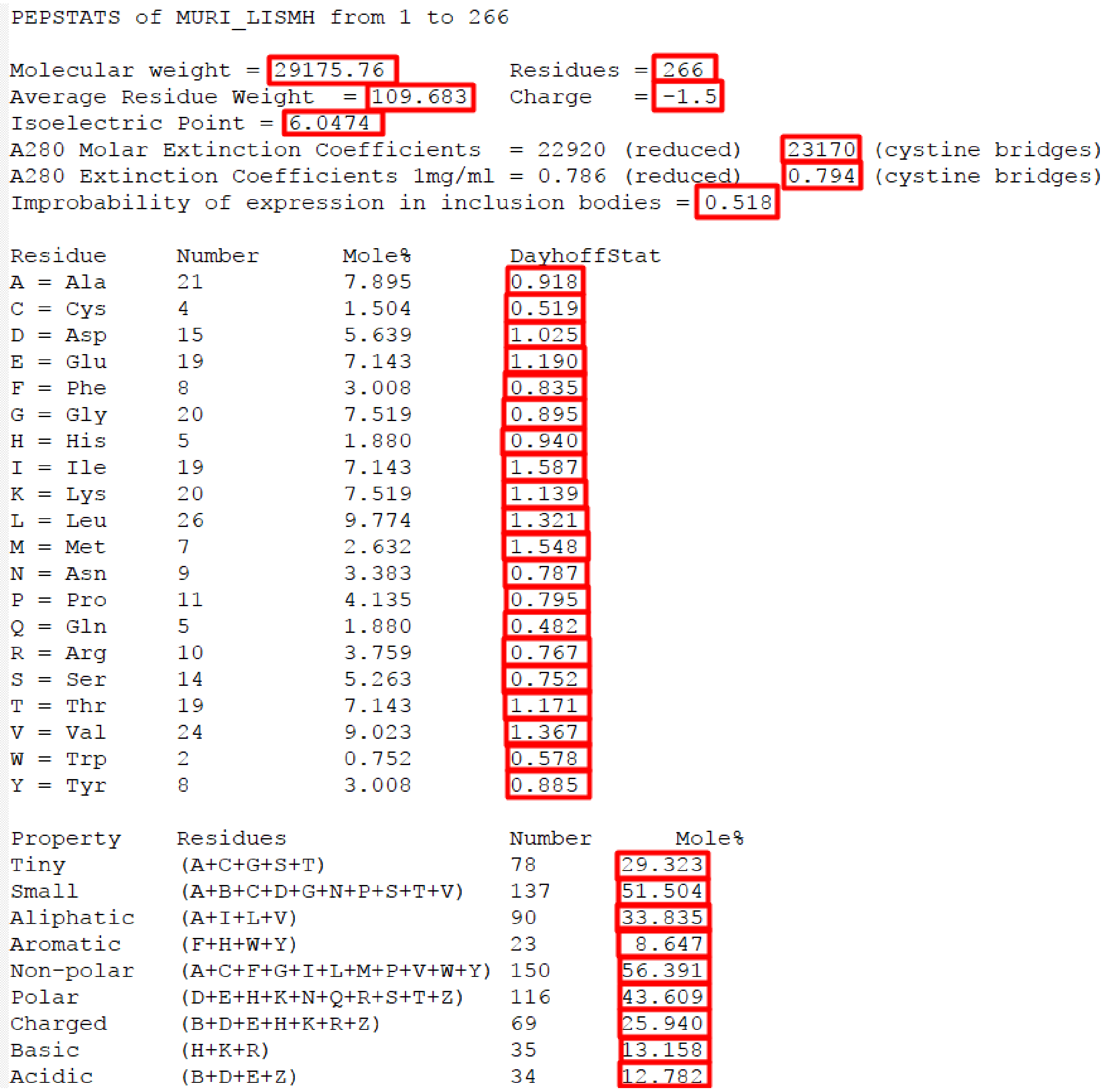
$$O_B = \frac{S_p - S_n}{S_p + S_n}, \tag{3.1}$$

where S_p is the sum of BLAST scores of proteins in the *k*-nearest neighbors in the positive training dataset. Similarly, S_n is the sum of scores of the *k*-nearest neighbor proteins in the negative training dataset. Note that the value of O_B is between -1 and +1. The output is 1 if all *k* nearest proteins are elements of the positive training dataset and -1 if all *k* proteins are from the negative training dataset.

3.2.2 PEPSTATS-SVM

The Pepstats tool [29] which is developed by European Molecular Biology Open Software Suite (EMBOSS) is used to extract the peptide statistics of the proteins. Each protein is represented by a 37-dimensional vector. Features that are used in 37-dimensional vector is shown in Figure 3.5. These features are scaled using LIBSVM

[35] and subsequently fed to the SVM classifier as input.



```
PEPSTATS of MURI_LISMH from 1 to 266

Molecular weight = 29175.76          Residues = 266
Average Residue Weight  = 109.683    Charge   = -1.5
Isoelectric Point = 6.0474
A280 Molar Extinction Coefficients  = 22920 (reduced)   23170 (cystine bridges)
A280 Extinction Coefficients 1mg/ml = 0.786 (reduced)   0.794 (cystine bridges)
Improbability of expression in inclusion bodies = 0.518

Residue         Number          Mole%           DayhoffStat
A = Ala         21              7.895           0.918
C = Cys         4               1.504           0.519
D = Asp         15              5.639           1.025
E = Glu         19              7.143           1.190
F = Phe         8               3.008           0.835
G = Gly         20              7.519           0.895
H = His         5               1.880           0.940
I = Ile         19              7.143           1.587
K = Lys         20              7.519           1.139
L = Leu         26              9.774           1.321
M = Met         7               2.632           1.548
N = Asn         9               3.383           0.787
P = Pro         11              4.135           0.795
Q = Gln         5               1.880           0.482
R = Arg         10              3.759           0.767
S = Ser         14              5.263           0.752
T = Thr         19              7.143           1.171
V = Val         24              9.023           1.367
W = Trp         2               0.752           0.578
Y = Tyr         8               3.008           0.885

Property        Residues                Number          Mole%
Tiny            (A+C+G+S+T)             78              29.323
Small           (A+B+C+D+G+N+P+S+T+V)   137             51.504
Aliphatic       (A+I+L+V)               90              33.835
Aromatic        (F+H+W+Y)               23               8.647
Non-polar       (A+C+F+G+I+L+M+P+V+W+Y) 150             56.391
Polar           (D+E+H+K+N+Q+R+S+T+Z)   116             43.609
Charged         (B+D+E+H+K+R+Z)         69              25.940
Basic           (H+K+R)                 35              13.158
Acidic          (B+D+E+Z)               34              12.782
```

Figure 3.5: Pepstats results for protein B8DHZ5 (MURI_LISMH). Totally, 37 peptide statistics are chosen for feature vector.

3.2.3 SPMap

Saraç, Gürsoy-Yüzügüllü, Cetin-Atalay and Atalay [31] previously developed a subsequence-based method to predict protein functions called Subsequence Profile Map (SPMap).

SPMap consists of two main parts: Subsequence Profile Map Construction and Feature Vector Generation. Flow diagram of SPMap is given in Figure 3.6.

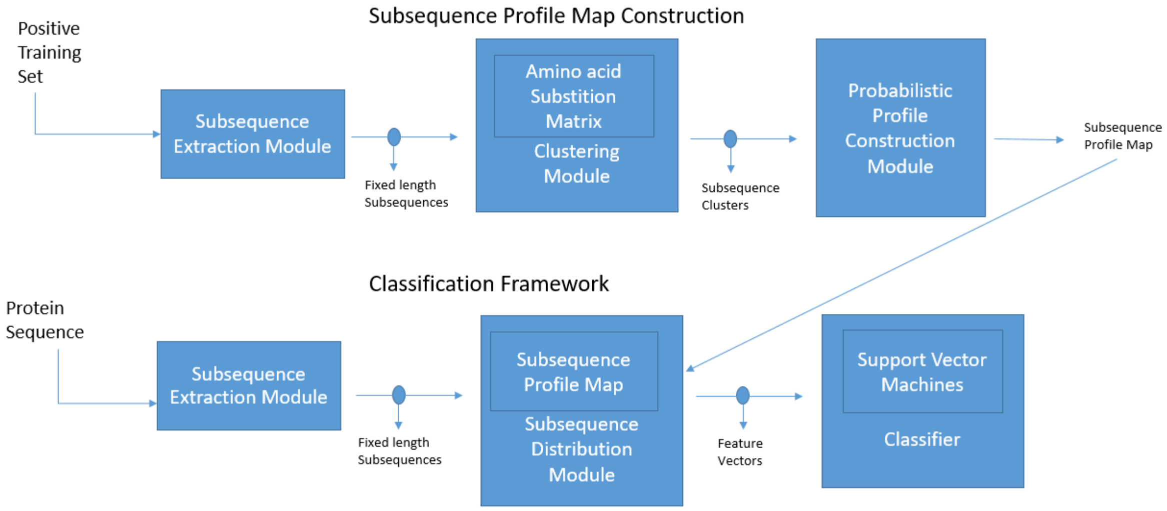

Subsequence Profile Map Construction
Positive Training Set
Subsequence Extraction Module
Fixed length Subsequences
Amino acid Substition Matrix
Clustering Module
Subsequence Clusters
Probabilistic Profile Construction Module
Subsequence Profile Map
Classification Framework
Protein Sequence
Subsequence Extraction Module
Fixed length Subsequences
Subsequence Profile Map
Subsequence Distribution Module
Feature Vectors
Support Vector Machines
Classifier

Figure 3.6: SPMap flow diagram.

3.2.3.1 Subsequence Profile Map Construction

Subsequence Profile Map Construction part consists of three modules:

- **Subsequence Extraction Module**

  All possible subsequences for given length l are extracted from the positive training dataset. Sliding window technique is used in order to extract all possible subsequences. For example, for a given string MSTNPKPQR, after extraction with l=5, all possible subsequences are obtained and they are:

  ```
  MSTNPKPQR
  MSTNP
   STNPK
    TNPKP
     NPKPQ
      PKPQR
  ```

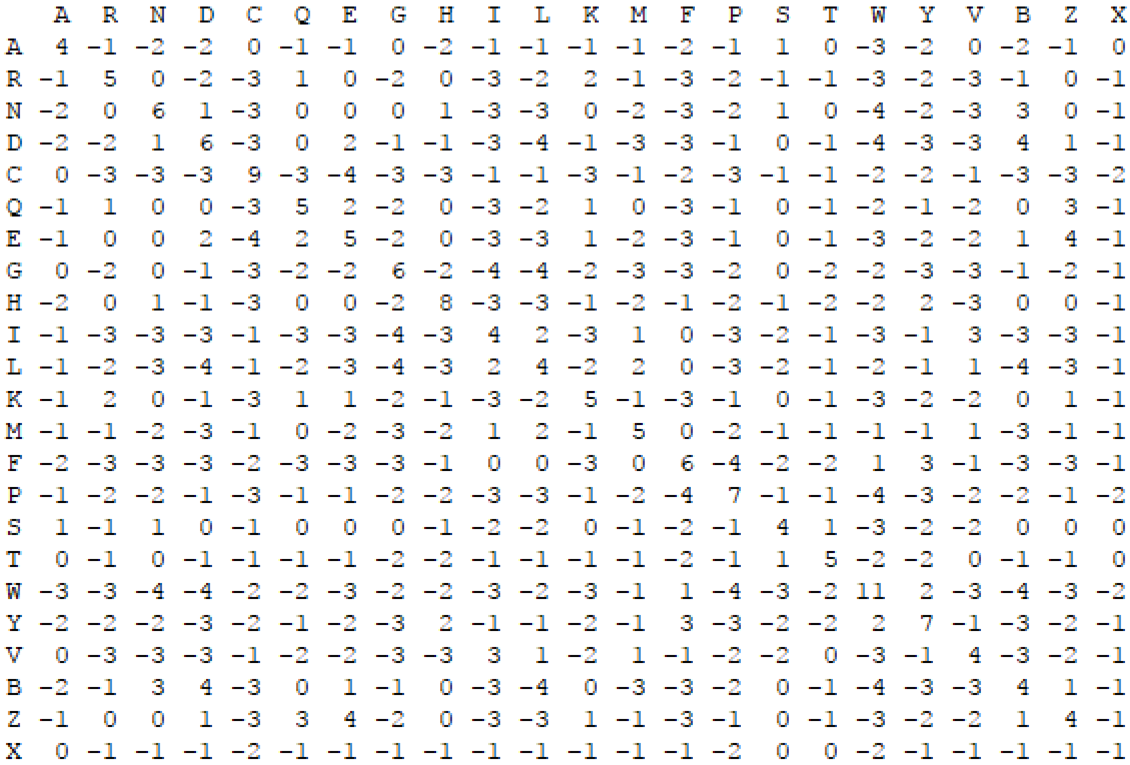
- **Clustering Module**

  After obtaining all possible subsequences, all subsequences are clustered based on their similarities. BLOcks SUbstitution Matrix (BLOSUM62) [36] is used to calculate similarity score between two subsequences. BLOSUM62, which is a substitution matrix, is used to align sequences and each entry represents a similarity score between two amino acids. BLOSUM62 is used to compute the similarity of two subsequences which is given in Figure 3.7. At a given instant of time, a subsequence is compared with all existing clusters and assigned to the cluster which gives the highest similarity score. Similarity score between two subsequences is calculated as follows:

  $$s(x, y) = \sum_{i=1}^{5} M(x(i), y(i)), \tag{3.2}$$

  where $x(i)$ is the i^{th} position of the amino acid x . $M(x(i), y(i))$ is the similarity score in BLOSUM62 matrix for the i^{th} position of x and y. For example, similarity score is calculated as follows for a given two subsequences x = MSTNP and y = STNPK,

  $$\begin{aligned} s(x, y) &= M(M, S) + M(S, T) + M(T, N) + M(N, P) + M(P, K) \\ &= (-1) + 1 + 0 + (-2) + -1 \\ &= -3 \end{aligned} \tag{3.3}$$



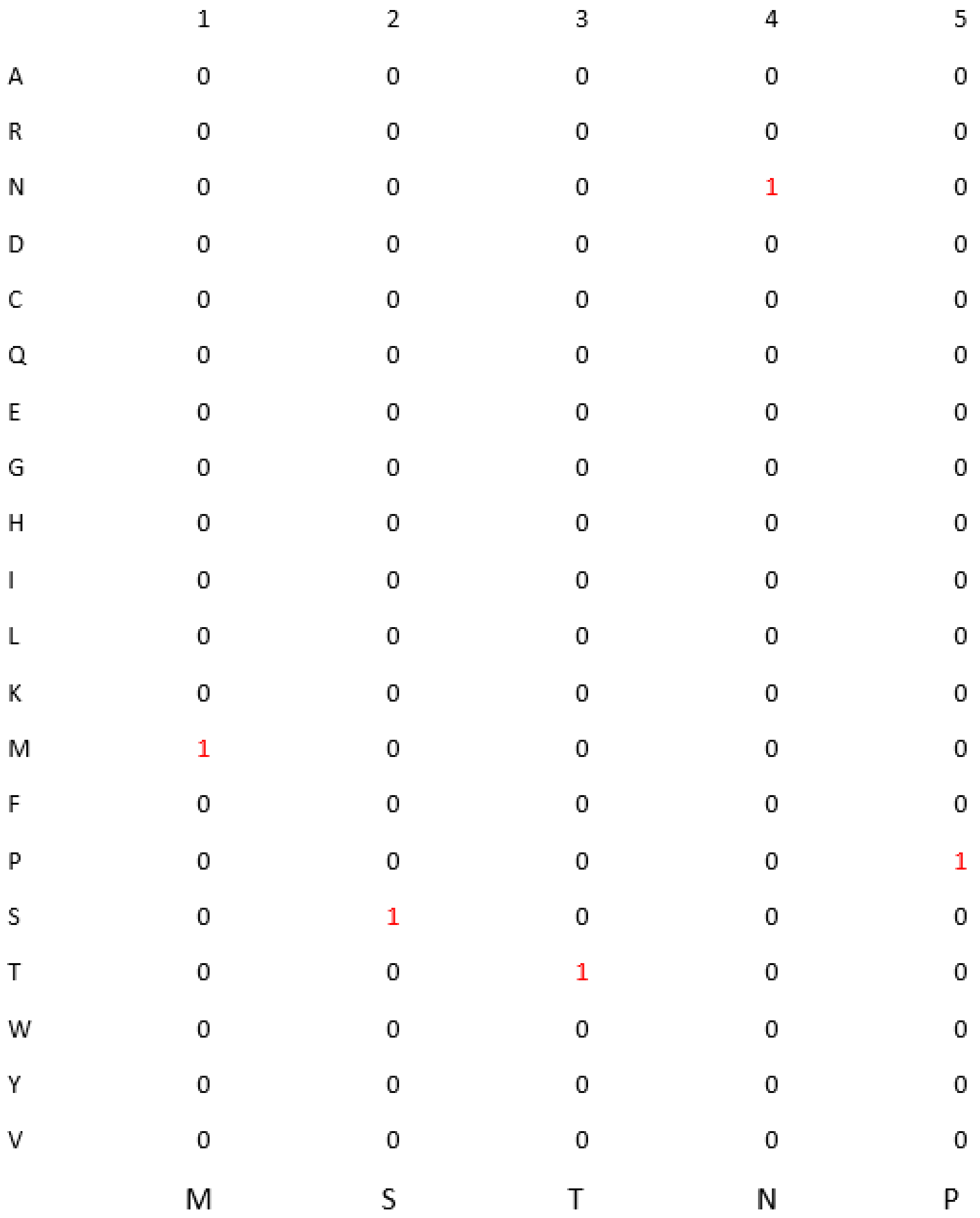
	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0	-2	-1	0
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3	-1	0	-1
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3	3	0	-1
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3	4	1	-1
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1	-3	-3	-2
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2	0	3	-1
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2	1	4	-1
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3	-1	-2	-1
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3	0	0	-1
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3	-3	-3	-1
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1	-4	-3	-1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2	0	1	-1
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1	-3	-1	-1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1	-3	-3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2	-2	-1	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2	0	0	0
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0	-1	-1	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3	-4	-3	-2
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1	-3	-2	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4	-3	-2	-1
B	-2	-1	3	4	-3	0	1	-1	0	-3	-4	0	-3	-3	-2	0	-1	-4	-3	-3	4	1	-1
Z	-1	0	0	1	-3	3	4	-2	0	-3	-3	1	-1	-3	-1	0	-1	-3	-2	-2	1	4	-1
X	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	0	0	-2	-1	-1	-1	-1	-1

Figure 3.7: Blosum62 matrix is used to calculate the similarity score between amino acids.

After calculating similarity score between a cluster c and a subsequence ss,

- If $s(c, ss) \geq 8$, the subsequence is assigned to this cluster.
- If $s(c, ss) < 8$, a new cluster is created.

After all clusters are generated, a position specific scoring matrix (PSSM) is created for each cluster which consists of 5 columns (l) and 20 rows (amino acid count). The amino acid count for each position is stored in the PSSM. Firstly, all columns on each row are initialized to 0. Then, the PSSM is updated according to the first subsequence. For a given subsequence MSTNP, M's count is incremented in the first position, S's count in the second position and T's count in the third position and so on. The first step of constructing PSSM is illustrated in Figure 3.8. PSSM then is updated using all subsequences belonging to that cluster.



	1	2	3	4	5
A	0	0	0	0	0
R	0	0	0	0	0
N	0	0	0	1	0
D	0	0	0	0	0
C	0	0	0	0	0
Q	0	0	0	0	0
E	0	0	0	0	0
G	0	0	0	0	0
H	0	0	0	0	0
I	0	0	0	0	0
L	0	0	0	0	0
K	0	0	0	0	0
M	1	0	0	0	0
F	0	0	0	0	0
P	0	0	0	0	1
S	0	1	0	0	0
T	0	0	1	0	0
W	0	0	0	0	0
Y	0	0	0	0	0
V	0	0	0	0	0
	M	S	T	N	P

Figure 3.8: First step of constructing Position Specific Scoring Matrix (PSSM). In each column, the associated amino acid count is incremented by 1.

- **Probabilistic Profile Construction Module**

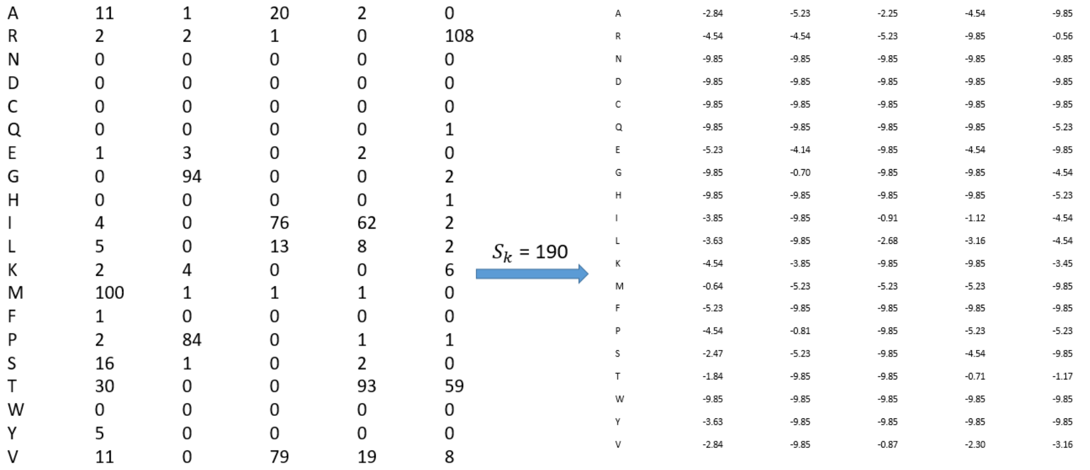
  After amino acid count is obtained with PSSM, each PSSM is converted to a probabilistic profile. S_k is the total number of subsequences for a cluster.

  - If the S_k is less than 10% of the positive training dataset size, that cluster is ignored as a profile.

- Otherwise, a probabilistic profile is generated. The probability of the amino acid j to occur at the i^{th} position of the subsequence is represented by $PP_k(i,j)$. The amino acid count for the amino acid j at the i^{th} position of the subsequence is represented by $Aa_{count}(i,j)$.

$$PP_k(i,j) = log\frac{Aa_{count}(i,j) + 0.01}{S_k}, \tag{3.4}$$

- Finally, the log of this result is taken and assigned to that position's amino acid. 0.01 is added to amino acid count for each position to avoid zero probabilities. Conversion of the PSSM to a probabilistic profile is given in Figure 3.9.



A	11	1	20	2	0
R	2	2	1	0	108
N	0	0	0	0	0
D	0	0	0	0	0
C	0	0	0	0	0
Q	0	0	0	0	1
E	1	3	0	2	0
G	0	94	0	0	2
H	0	0	0	0	1
I	4	0	76	62	2
L	5	0	13	8	2
K	2	4	0	0	6
M	100	1	1	1	0
F	1	0	0	0	0
P	2	84	0	1	1
S	16	1	0	2	0
T	30	0	0	93	59
W	0	0	0	0	0
Y	5	0	0	0	0
V	11	0	79	19	8

S_k = 190

A	-2.84	-5.23	-2.25	-4.54	-9.85
R	-4.54	-4.54	-5.23	-9.85	-0.56
N	-9.85	-9.85	-9.85	-9.85	-9.85
D	-9.85	-9.85	-9.85	-9.85	-9.85
C	-9.85	-9.85	-9.85	-9.85	-9.85
Q	-9.85	-9.85	-9.85	-9.85	-5.23
E	-5.23	-4.14	-9.85	-4.54	-9.85
G	-9.85	-0.70	-9.85	-9.85	-4.54
H	-9.85	-9.85	-9.85	-9.85	-5.23
I	-3.85	-9.85	-0.91	-1.12	-4.54
L	-3.63	-9.85	-2.68	-3.16	-4.54
K	-4.54	-3.85	-9.85	-9.85	-3.45
M	-0.64	-5.23	-5.23	-5.23	-9.85
F	-5.23	-9.85	-9.85	-9.85	-9.85
P	-4.54	-0.81	-9.85	-5.23	-5.23
S	-2.47	-5.23	-9.85	-4.54	-9.85
T	-1.84	-9.85	-9.85	-0.71	-1.17
W	-9.85	-9.85	-9.85	-9.85	-9.85
Y	-3.63	-9.85	-9.85	-9.85	-9.85
V	-2.84	-9.85	-0.87	-2.30	-3.16

Figure 3.9: Converting the PSSM to a probabilistic profile.

3.2.3.2 Feature Vector Generation

We are looking for the probability of a subsequence to be generated by a profile and the highest probability value is sought among the profiles for a subsequence. The highest probability value is used as a member of the feature vector for that profile. When we do this for all subsequences, the contribution of each cluster is determined for the input sequence. A feature vector is calculated as follows, firstly, each subsequence ss is compared with all existing probabilistic profiles PP_k and a probability is computed as

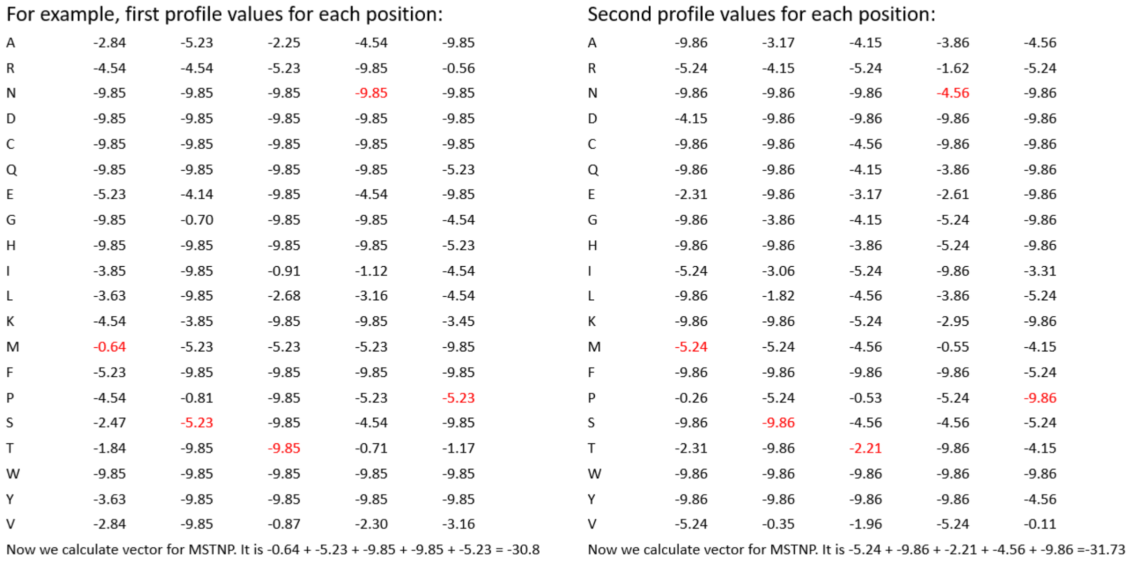
$$P(ss|PP_k) = \sum_{i=1}^{5} PP_k(i, ss(i)), \qquad (3.5)$$

The element of j^{th} dimension of the feature vector V is determined as

$$V(j) = max_{ss_i \in E} P(ss_i|PP_k), \qquad (3.6)$$

The probability of subsequence with the highest score of protein E on PP_k.

For example, for the subsequence MSTNP, all the elements of a profile are visited. For each element of MSTNP the score of each amino acid is obtained and the sum of these scores are calculated. Feature vector generation is shown in Figure 3.10.



For example, first profile values for each position:

A	-2.84	-5.23	-2.25	-4.54	-9.85
R	-4.54	-4.54	-5.23	-9.85	-0.56
N	-9.85	-9.85	-9.85	-9.85	-9.85
D	-9.85	-9.85	-9.85	-9.85	-9.85
C	-9.85	-9.85	-9.85	-9.85	-9.85
Q	-9.85	-9.85	-9.85	-9.85	-5.23
E	-5.23	-4.14	-9.85	-4.54	-9.85
G	-9.85	-0.70	-9.85	-9.85	-4.54
H	-9.85	-9.85	-9.85	-9.85	-5.23
I	-3.85	-9.85	-0.91	-1.12	-4.54
L	-3.63	-9.85	-2.68	-3.16	-4.54
K	-4.54	-3.85	-9.85	-9.85	-3.45
M	-0.64	-5.23	-5.23	-5.23	-9.85
F	-5.23	-9.85	-9.85	-9.85	-9.85
P	-4.54	-0.81	-9.85	-5.23	-5.23
S	-2.47	-5.23	-9.85	-4.54	-9.85
T	-1.84	-9.85	-9.85	-0.71	-1.17
W	-9.85	-9.85	-9.85	-9.85	-9.85
Y	-3.63	-9.85	-9.85	-9.85	-9.85
V	-2.84	-9.85	-0.87	-2.30	-3.16

Now we calculate vector for MSTNP. It is -0.64 + -5.23 + -9.85 + -9.85 + -5.23 = -30.8

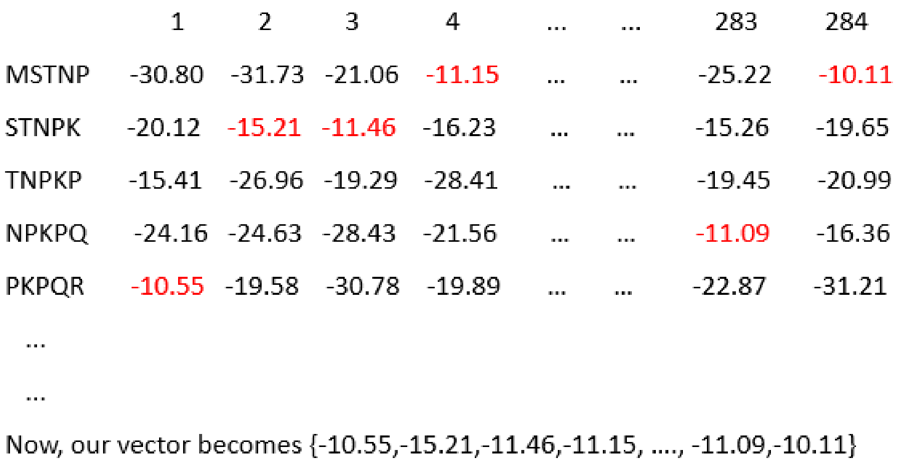
Second profile values for each position:

A	-9.86	-3.17	-4.15	-3.86	-4.56
R	-5.24	-4.15	-5.24	-1.62	-5.24
N	-9.86	-9.86	-9.86	-4.56	-9.86
D	-4.15	-9.86	-9.86	-9.86	-9.86
C	-9.86	-9.86	-4.56	-9.86	-9.86
Q	-9.86	-9.86	-4.15	-3.86	-9.86
E	-2.31	-9.86	-3.17	-2.61	-9.86
G	-9.86	-3.86	-4.15	-5.24	-9.86
H	-9.86	-9.86	-3.86	-5.24	-9.86
I	-5.24	-3.06	-5.24	-9.86	-3.31
L	-9.86	-1.82	-4.56	-3.86	-5.24
K	-9.86	-9.86	-5.24	-2.95	-9.86
M	-5.24	-5.24	-4.56	-0.55	-4.15
F	-9.86	-9.86	-9.86	-9.86	-5.24
P	-0.26	-5.24	-0.53	-5.24	-9.86
S	-9.86	-9.86	-4.56	-4.56	-5.24
T	-2.31	-9.86	-2.21	-9.86	-4.15
W	-9.86	-9.86	-9.86	-9.86	-9.86
Y	-9.86	-9.86	-9.86	-9.86	-4.56
V	-5.24	-0.35	-1.96	-5.24	-0.11

Now we calculate vector for MSTNP. It is -5.24 + -9.86 + -2.21 + -4.56 + -9.86 =-31.73

Figure 3.10: Feature vector generation. Numbers on each column with color red denotes the score of the amino acid for that position. All these scores are summed for feature vector generation.

For each profile, the same operation is applied and a feature vector is generated. Then, with each coming new subsequence another vector is generated. And all these vectors for each profile are compared and the highest score for each profile is selected. Constructing a feature generation is illustrated in Figure 3.11.



	1	2	3	4	...	...	283	284
MSTNP	-30.80	-31.73	-21.06	-11.15	...	...	-25.22	-10.11
STNPK	-20.12	-15.21	-11.46	-16.23	...	...	-15.26	-19.65
TNPKP	-15.41	-26.96	-19.29	-28.41	...	...	-19.45	-20.99
NPKPQ	-24.16	-24.63	-28.43	-21.56	...	...	-11.09	-16.36
PKPQR	-10.55	-19.58	-30.78	-19.89	...	...	-22.87	-31.21
...								
...								

Now, our vector becomes {-10.55,-15.21,-11.46,-11.15, ..., -11.09,-10.11}

Figure 3.11: Constructing a feature vector. The scores with color red are selected for feature vector since their score is the highest score for that position. Each number on columns represent the probabilistic profile. The first element of each row is one of the possible subsequences.

After processing all subsequences, a feature vector corresponding to the input protein sequence is generated. Each element of the vector is changed back its natural logarithms (between 0 and 1), using exp function. For each positive and negative proteins same operations are applied and finally, a training file is created. SVM^{light} [37] is used as a classifier.

3.2.4 Combining Ensemble Methods

5-fold cross-validation is applied for each method and area under the Receiver Operating Characteristic (ROC) score is calculated for BLAST-*k*NN, PEPSTSTATS-SVM and SPMap. Using these ROC scores all three methods are combined and weighted mean score for each method is calculated. Weighted mean score for method m where $m \in \{BLAST - kNN, PEPSTSTATS - SVM, SPMap\}$ is calculated as follows;

$$W(m) = \frac{R_m^4}{R_{BLAST-kNN}^4 + R_{SPMap}^4 + R_{PEPSTATS-SVM}^4}, \qquad (3.7)$$

where the weighted mean score for method m is represented by $W(m)$, where m can be either one of the methods that we applied. R_m stands for ROC score for method m.

In order to see the effect of each method these weighted means scores are calculated. For a target protein, a prediction score is given by each method and these prediction scores are then multiplied by weighted means of methods and a final prediction score is obtained by their sum. The method which has the highest weighted mean score affects mostly the final score.

3.3 Determining The Optimal Cut-off Value for EC Classes

In this section, positive and negative cut-off values are determined and these cut-off values are used to decide whether a given protein sequence is either positive or negative prediction. Determining cut-off value for EC classes is one of the crucial tasks in this study. Since each EC class trained with its own data, cut-off values are calculated for each EC class, separately. F-score statistics are used for calculating the optimal cut-off value. F-score statistics are commonly employed in binary classification problems. The harmonic mean between precision and recall is the F-score and F-score measure calculated as follows,

$$\begin{aligned} Precision &= \frac{TP}{TP+FP}, \\ Recall &= \frac{TP}{TP+FN}, \\ F-score &= \frac{2 \times Precision \times Recall}{Precision + Recall}. \end{aligned} \tag{3.8}$$

TP, FP, TN, and FN denotes true positive, false positive, true negative and false negative, respectively. Positive and negative cut-off values are calculated separately to decide whether a given input sequence is either positive or negative prediction.

3.3.1 Determining Positive Optimal Cut-off Values for EC Classes

Positive and negative validation sets are used to determine positive optimal cut-off values. Constructing positive validation sets are explained in Section 3.1.1, Section 3.1.2 and Section 3.1.3. Pseudocode of determining positive cut-off values are given in Algorithm 1. If a protein from positive validation set gets prediction score above the cut-off value, it is labeled as true positive. Otherwise, it is labeled as false negative. Furthermore, if a protein from negative validation set gets prediction score above the cut-off value, it is labeled as false positive. Otherwise, it is labeled as true negative. After obtaining true positive, false negative, false positive and true negative precision, recall and F-score values are calculated. The maximum cut-off value value which gives the highest F-score is selected as the positive cut-off value for that EC class. However, some of these cut-off values are too close to 1.0 so we decided to decrease these values using the rules in given Algorithm 2. For each EC class which has F-score greater than 0.9 and has cut-off value greater than 0.9 the minimum and the maximum cut-off values are determined. Subsequently, the average of the minimum and the maximum cut-off values is calculated and determined as the positive cut-off value for that EC class. Finally, in order to avoid false positives cut-off values, less than 0.6 are fixed to 0.6.

3.3.2 Determining Negative Optimal Cut-off Values for EC Classes

Positive and negative validation sets are also used to determine negative optimal cut-off values. Pseudocode of determining negative optimal cut-off values is given in Algorithm 3. If a protein from positive validation set gets prediction score above the cut-off value, it is labeled as true negative. Otherwise, it is labeled as false positive. Furthermore, if a protein from negative validation set gets prediction score above the cut-off value, it is labeled as false negative. Otherwise, it is labeled as true positive. After obtaining true positive, false negative, false positive and true negative precision, recall and F-score values are calculated. The minimum cut-off value which gives the highest F-score is selected as the negative optimal cut-off value for that EC class. After investigating negative F-score results for all EC classes, we found out that highest F-scores are obtained between 0.25 and 0.35 so we decided to use cut-off value 0.3 as a global negative cut-off value for all EC classes. As an example, negative cut-off values and their F-score values for EC class 1.1.1.94 are given in Table 3.7.

Algorithm 1 Pseudocode of Determining Positive Cut-off Values for EC Classes

```
Require: $positive\_test\_proteins$: is a map where keys are EC Classes, values are proteins from positive validation set of corresponding EC classes, $negative\_test\_proteins$: is a map where keys are EC classes and values are proteins from negative validation set of corresponding EC classes, $class\_list$: list of trained EC classes
  for all $class \in class\_list$ do
    $cutoff \leftarrow 1.00$
    while $cutoff \geq 0.0$ do
      for all $pos\_P \in positive\_test\_proteins[class]$ do
        $PSCORE$ holds prediction score of $class$ for $pos\_P$
        if $PSCORE \geq cutoff$ then
          $TP \leftarrow TP + 1$
        else
          $FN \leftarrow FN + 1$
        end if
      end for
      for all $neg\_P \in negative\_test\_proteins[class]$ do
        $PSCORE$ holds prediction score of $class$ for $neg\_P$
        if $PSCORE \geq cutoff$ then
          $FP \leftarrow FP + 1$
        else
          $TN \leftarrow TN + 1$
        end if
      end for
      $cutoff \leftarrow cutoff - 0.011$
    end while
    Calculate $F - score$ according to the Equation 3.8
  end for
```

Algorithm 2 Pseudocode of Determining Optimal Positive Cut-off Values for EC Classes

```
Require: $positive\_fscores$: is a map where keys are cut-off values, values are F-scores from positive F-score results, $class\_list$: list of trained EC classes
  for all $class \in class\_list$ do
    $optimal\_cutoff \leftarrow 0.00, min\_cutoff \leftarrow 1.00, max\_cutoff \leftarrow 0.00$
    $max\_fscore \leftarrow 0.00, cutoff \leftarrow 1.00, final\_fscore \leftarrow 0.00$
    $final\_cutoff \leftarrow 0.00$
    while $cutoff \geq 0.0$ do
      $FScore$ holds F-score of current cut-off value
      if $FSCORE \geq max\_cutoff$ then
        $max\_cutoff \leftarrow FSCORE$
        $optimal\_cutoff \leftarrow cutoff$
      end if
      if $FSCORE \geq 0.90$ and $cutoff \leq min\_cutoff$ then
        $min\_cutoff \leftarrow cutoff$
      else if $FSCORE \geq 0.90$ and $cutoff \geq max\_cutoff$ then
        $max\_cutoff \leftarrow cutoff$
      end if
      $cutoff \leftarrow cutoff - 0.01$
    end while
    if $optimal\_cutoff \geq 0.90$ and $max\_cutoff \geq 0.90$ then
      $final\_cutoff \leftarrow (max\_cutoff + min\_cutoff)/2$
      $final\_fscore \leftarrow$ fscore of final_cutoff
    else
      $final\_cutoff \leftarrow optimal\_cutoff$
      $final\_fscore \leftarrow max\_fscore$
    end if
    if $final\_cutoff < 0.60$ then
      $final\_cutoff \leftarrow 0.6$
      $final\_fscore \leftarrow$ fscore of 0.6
    end if
  end for
```

Algorithm 3 Pseudocode of Determining Negative Cut-off Values for EC Classes

```
Require: $positive\_test\_proteins$: is a map where keys are EC Classes, values are proteins from positive validation set of corresponding EC classes, $negative\_test\_proteins$: is a map where keys are EC classes and values are proteins from negative validation set of corresponding EC classes, $class\_list$: list of trained EC classes
  for all $class \in class\_list$ do
    $cutoff \leftarrow 1.00$
    while $cutoff \geq 0.0$ do
      for all $pos\_P \in positive\_test\_proteins[class]$ do
        $PSCORE$ holds prediction score of $class$ for $pos\_P$
        if $PSCORE \geq cutoff$ then
          $TN \leftarrow TN + 1$
        else
          $FP \leftarrow FP + 1$
        end if
      end for
      for all $neg\_P \in negative\_test\_proteins[class]$ do
        $PSCORE$ holds prediction score of $class$ for $neg\_P$
        if $PSCORE \geq cutoff$ then
          $FN \leftarrow FN + 1$
        else
          $TP \leftarrow TP + 1$
        end if
      end for
      $cutoff \leftarrow cutoff - 0.011$
    end while
    Calculate $F - score$ according to the Equation 3.8
  end for
```

Table 3.7: Negative cut-off values and their F-score values for EC class 1.1.1.94.

Cut-off Value	**F-score**
0.15	0.99
0.16	0.99
0.17	0.993
0.18	0.995
0.19	0.995
0.20	0.995
0.21	0.995
0.22	0.995
0.23	0.995
0.24	0.997
0.25	0.997
0.26	0.998
0.27	0.998
0.28	0.998
0.29	0.998
0.30	1.00
0.31	1.00
0.32	1.00
0.33	1.00
0.34	1.00
0.35	1.00

3.4 Flowchart of ECPred

The flowchart of ECPred is given in Figure 3.12. For a given query sequence, firstly we decide whether it is an enzyme or non-enzyme. if it is an enzyme, then its main class is determined. Subsequently, the query sequence is input to the Level 2 classifiers. The EC subfamily class with the highest prediction score is determined. If the highest prediction score is greater than the threshold of that EC subfamily class, the query sequence is labeled with the EC subfamily class, otherwise, the algorithm stops. For the Level 3 and Level 4 classifiers, same operations are applied. If the prediction score is greater than the threshold, predictions are continued, otherwise, predictions are stopped.

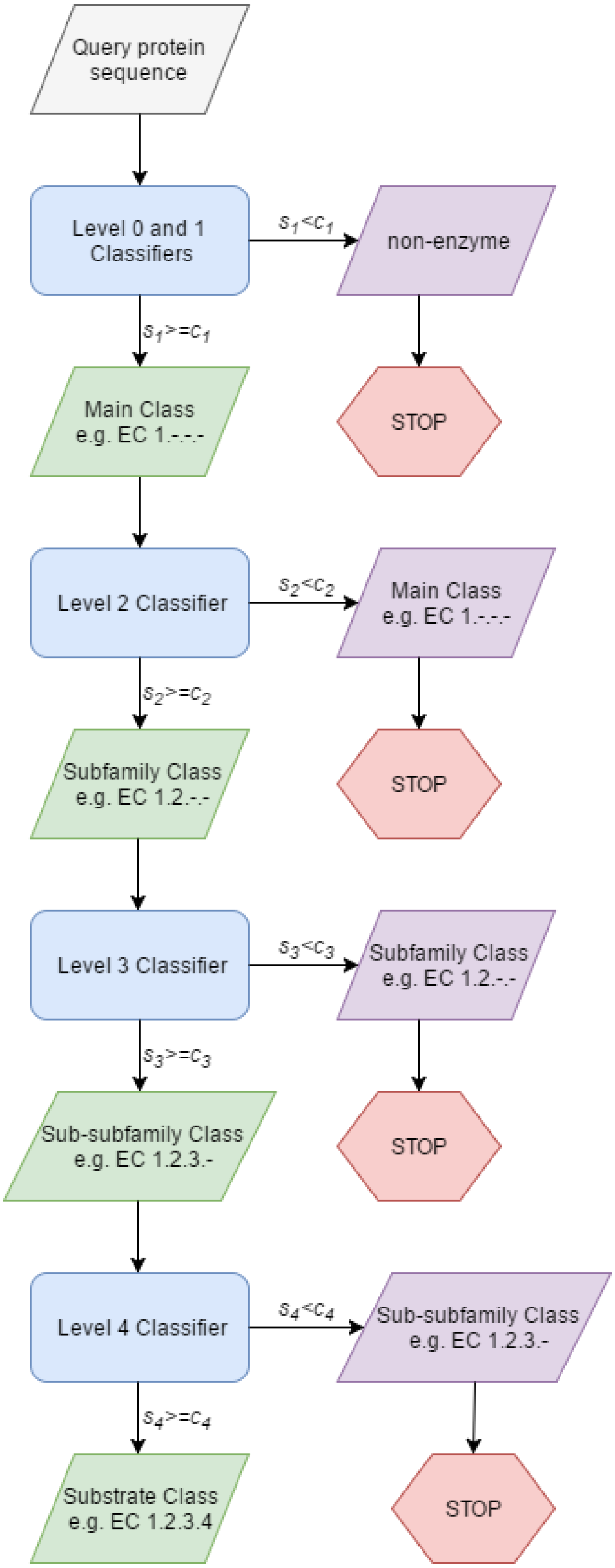

Query protein sequence
Level 0 and 1 Classifiers
s1<c1
non-enzyme
s1>=c1
Main Class e.g. EC 1.-.-.-
STOP
Level 2 Classifier
s2<c2
Main Class e.g. EC 1.-.-.-
s2>=c2
Subfamily Class e.g. EC 1.2.-.-
STOP
Level 3 Classifier
s3<c3
Subfamily Class e.g. EC 1.2.-.-
s3>=c3
Sub-subfamily Class e.g. EC 1.2.3.-
STOP
Level 4 Classifier
s4<c4
Sub-subfamily Class e.g. EC 1.2.3.-
s4>=c4
Substrate Class e.g. EC 1.2.3.4
STOP

Figure 3.12: The flowchart of ECPred. For a given query sequence, if its prediction score s_m greater than cut-off value c_m where m is EC hierarchy Level, prediction is continued otherwise prediction is stopped.

CHAPTER 4

RESULTS AND DISCUSSION

In this study, 6 main class classifiers, 55 subfamily classifiers, 163 sub-subfamily classifiers and 634 substrate classifiers are trained. Performance measurement of the classifiers is done using F-score statistics which is explained in Section 3.4.

In this chapter, Level 0-4 performance results are examined in Section 4.1 and 4.2. Performance results of proteins that do not have functional domain information are given in Section 4.3. *ECPred* is compared with three web-based tools and the performance results are given in Section 4.4.

4.1 Level 0 and Level 1 Results

4.1.1 Enzyme/non-enzyme and Main Class Results

In order to determine the optimal positive threshold for six main EC classes, F-score is calculated using the validation set. For each class, the protein sequences from the other five main classes and the non-enzymes are selected in the negative validation set. The positive F-score is calculated in order to determine positive cut-off values for six main EC classes. The positive F-score plot for six main EC classes is given in Figure 4.1. An average F-score value of 0.91 is obtained for six main EC classes. Only an F-score which is less than 0.9 is obtained for Lysases. For each main EC class, a cut-off value between 0 and 1 is selected that gives the highest F-score. Cut-off values for six main EC classes are obtained as follows: 0.59, 0.52, 0.52, 0.63, 0.58 and 0.64 for Oxidoreductases, Transferases, Hydrolases, Lyases, Isomerases and

Ligases, respectively.

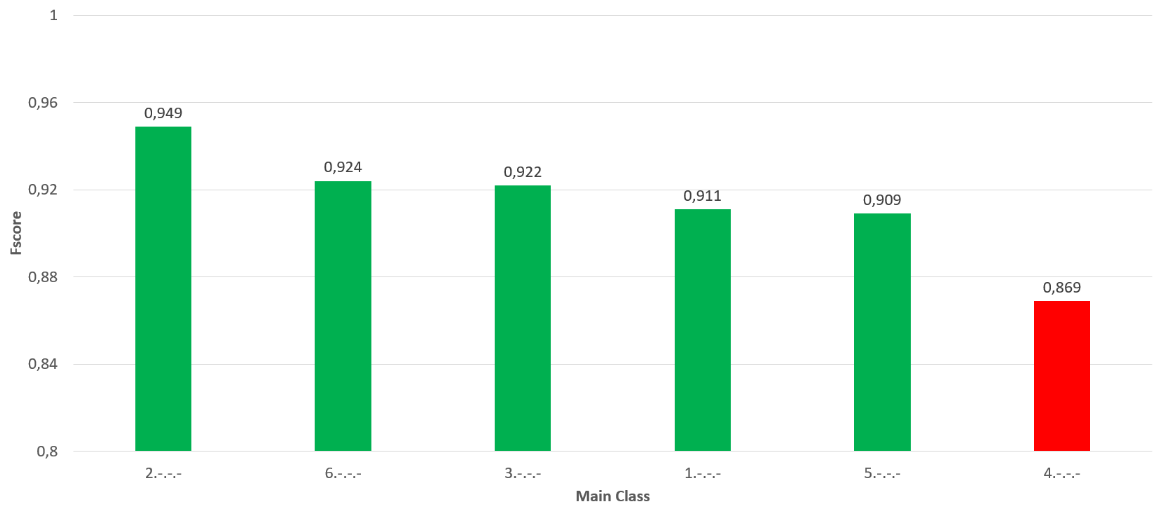


Figure 4.1: Plot of main classes versus their positive F-scores. Green color means that F-score is greater than 0.9. Red color means that F-score is less than 0.9.

Using *ECPred* we tried to predict which functions that a protein doesn't have. Non-enzymes are a specific example of these functions. To do this, negative F-scores are calculated for six main EC classes and their results are given in Figure 4.2. In the calculation of the negative F-scores, non-enzymes are treated as positives and marked as true positive or false negative. On the contrary, main class proteins are treated as negatives and marked as true negative and false positive. Results show that six main EC classes can classify non-enzymes with average F-score of 0.98, individually. However, six main classifiers give low prediction score at the same time with F-score of 0.93 which is the actual performance result of *ECPred* for the non-enzyme classification.

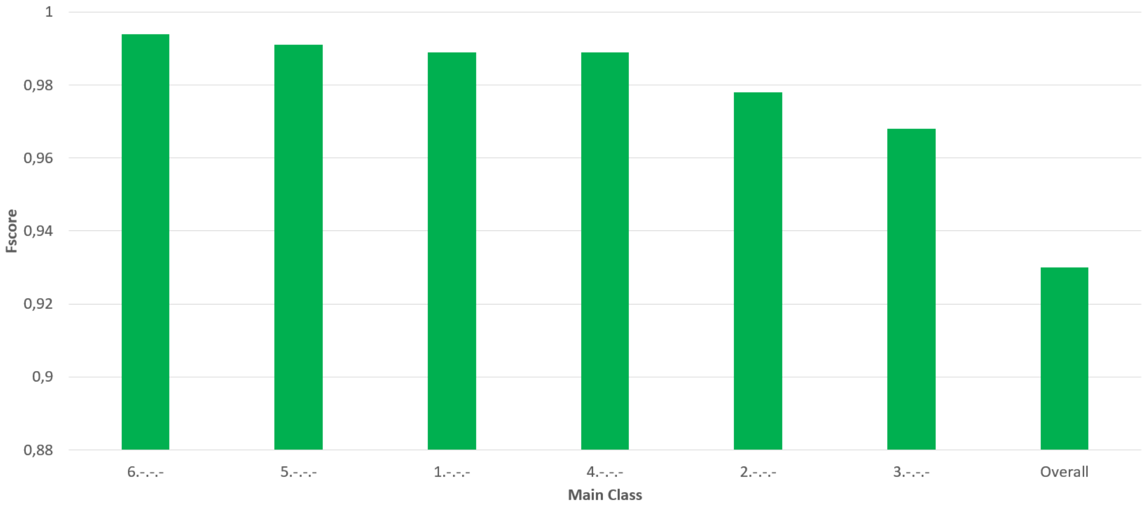


Figure 4.2: Plot of main classes versus their negative F-scores. Overall label indicates that six main classifiers give low prediction score at the same time.

4.2 Level 2, Level 3 and Level 4 Results

4.2.1 Subfamily Class Results

61 EC subfamily classes are trained and the average positive F-score is calculated as 0.98. The F-score plot for subfamily classes is given in Figure 4.3. Only F-score values lower than 0.9 are obtained for EC subfamily classes of 1.2. and 3.7. Results show that *ECPred* can predict sub-family classes with a high performance.

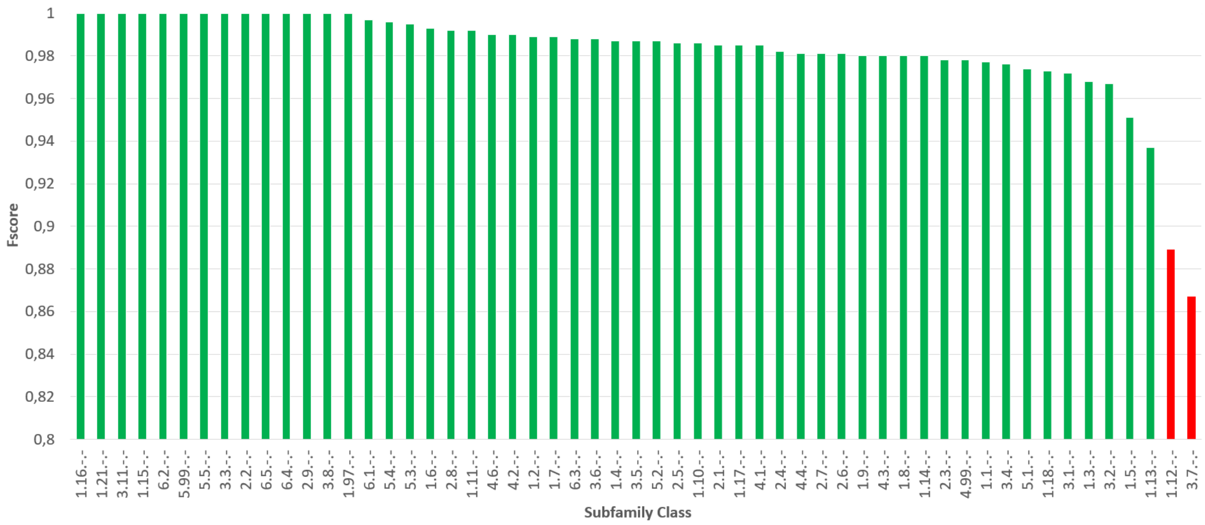


Figure 4.3: Plot of subfamily classes versus their F-scores. Green color means that F-score is greater than 0.9 while red color indicates an F-score less than 0.9.

4.2.2 Sub-subfamily Class Results

Totally, 163 EC sub-subfamily class is trained and average positive F-score is calculated as 0.98. The F-score plot for sub-subfamily classes is given in Figure 4.4. Except for two sub-subfamily classes, F-score values are higher than 0.9 for all sub-subfamily classes.

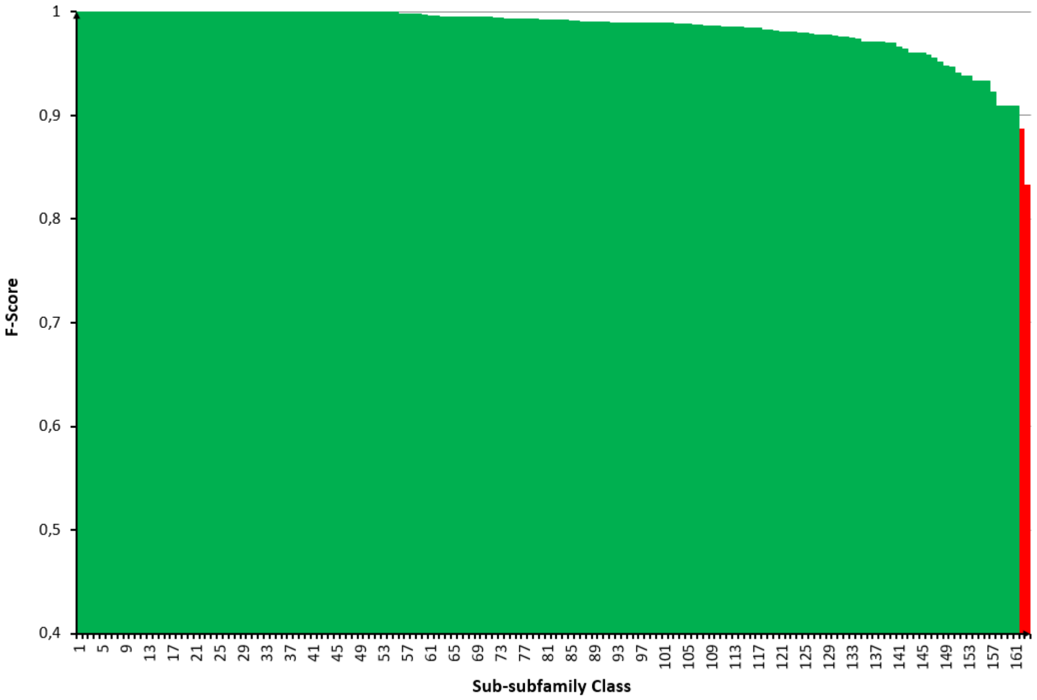


Figure 4.4: Plot of sub-subfamily classes versus their F-scores. Green color means that F-score is greater than 0.9 while red color indicates an F-score less than 0.9.

4.2.3 Substrate Class Results

Totally, 634 EC substrate class is trained and average positive F-score is calculated as 0.99. The F-score plot for substrate classes is given in Figure 4.5. About 90% of the substrate classes get F-score value of 1 and only for 5 EC classes F-score values are less than 0.9. Results show that *ECPred* can classify substrate classes with perfect performance.

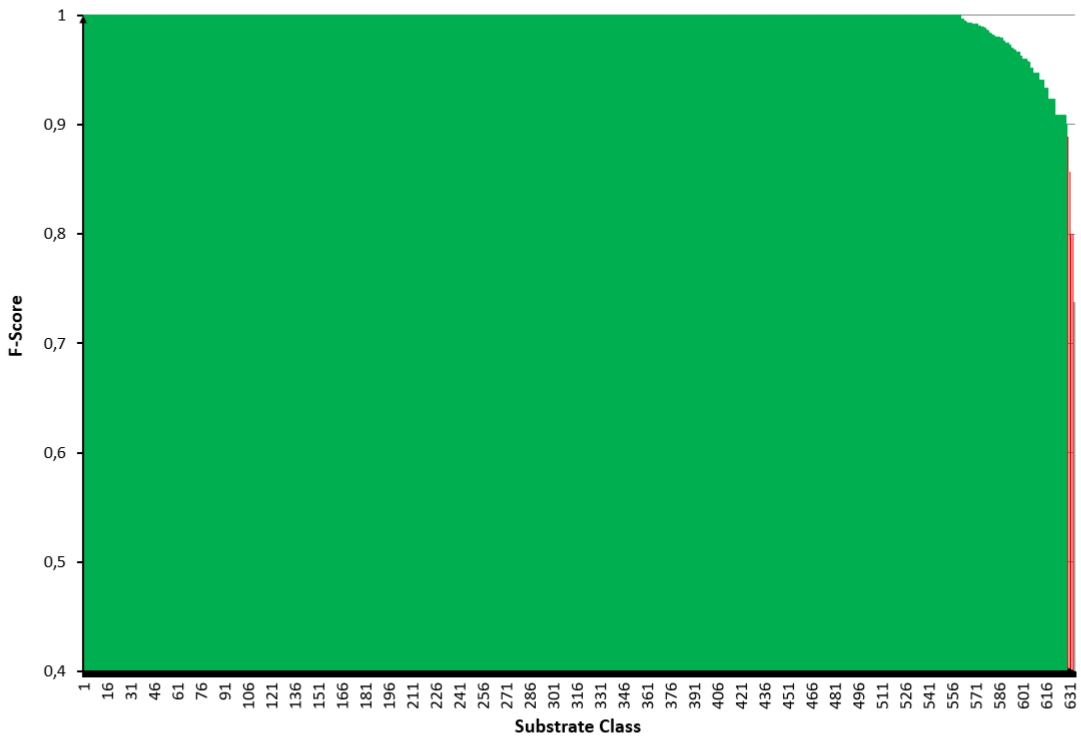


Figure 4.5: Plot of substrate classes versus their F-scores. Green color means that F-score is greater than 0.9 while red color indicates an F-score less than 0.9.

4.3 Protein Based Performance Results

Protein based performance is calculated for each EC Level. The results are given in Table 4.1. Total number of TP, FN, TN and FP are calculated for proteins at a level and precision, recall and F-score values are calculated according to these values. When we go deeper at EC hierarchy, results are getting better since the subclasses are more specific than their parents; therefore, the proteins in each class have more common features.

4.4 Individual vs. Combined Classifiers

The comparison between individual classifiers and their combination is made in this part of the study. The maximum F-scores are calculated for BLAST-*k*NN, SPMap

Table 4.1: Protein based performance results.

EC Level	Precision	Recall	F-score
Level 0	0.96	0.96	0.96
Level 1	0.96	0.90	0.93
Level 2	1.00	0.97	0.98
Level 3	1.00	0.98	0.99
Level 4	1.00	0.99	1.00
Average	1.00	0.97	0.98

and Pepstats-SVM. F-score values for trained 858 EC classes are sorted in descending order for three classifiers and the combined classifier. The plot for individual and combined classifiers is given in Figure 4.6. The results show that combined classifier performance results are better than three methods performance in most EC classes, however, BLAST-*k*NN performance is better than the combined classifier performance in a few number of EC classes. Additionally, performances of BLAST-*k*NN and SPMap are better than Pepstats performance.

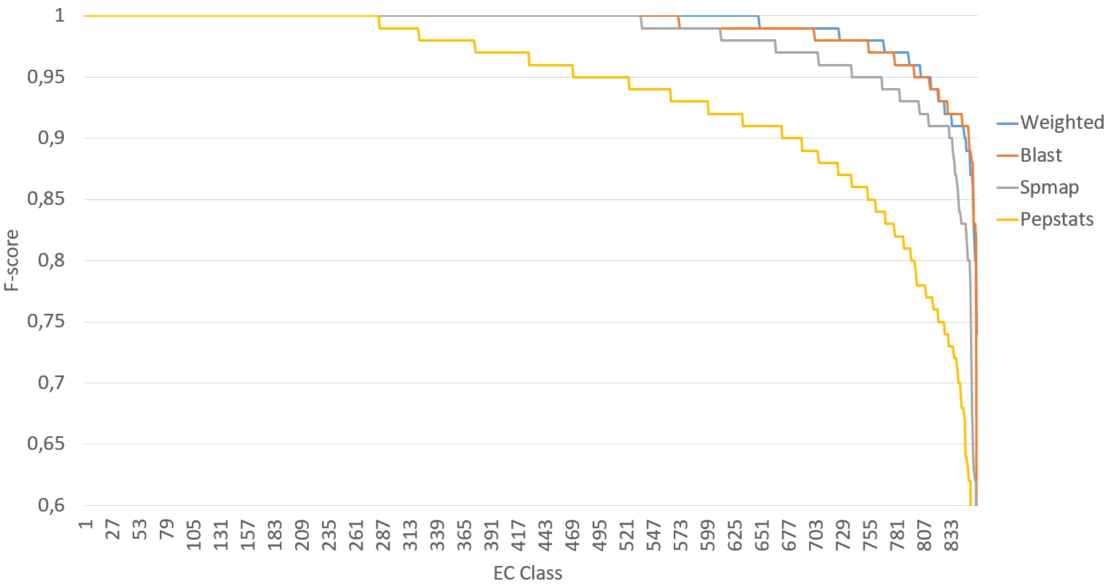


Figure 4.6: Performance results of individual and combined classifiers for 858 EC classes.

4.5 Weights of Three Independent Classifiers

In this part of the study, weights of BLAST-*k*NN, SPMap and Pepstats-SVM classifiers are plotted. The plot for weights of three independent EC classifiers based on BLAST-*k*NN for all EC classes is given in Figure 4.7. The results show that weights of BLAST-*k*NN and SPMap are parallel to each other and weights of both methods are higher than weight of Pepstats. The plots for weights of three independent EC classifiers based on SPMap and Pepstats for all EC classes are given in Figure 4.7. These results are also show that weights of BLAST and SPMap are close to each other and weight of Pepstats-SVM is less than weights of these two methods.

The plot for weights of three independent EC classifiers for six main EC classes is given in Figure 4.10. As it can be seen from figure, weight of BLAST-*k*NN is higher than weights of other two methods and weight of SPMap is greater than weight of Pepstats. The plots for weights of three independent EC classifiers for subfamily and sub-subfamily classes are given in Figure 4.11, Figure 4.12. The results show that weights of SPMap and BLAST-*k*NN are very similar to each other and weights of both methods are higher than weight of Pepstats-SVM. Finally, the plot for weights of three independent EC classifiers for six main EC classes is given in Figure 4.13. As it can be seen from figure, for most of the EC classes weights of three methods are close to each other. The plots are drown based on weight of BLAST-*k*NN for Figure 4.11, Figure 4.12 and Figure 4.13.

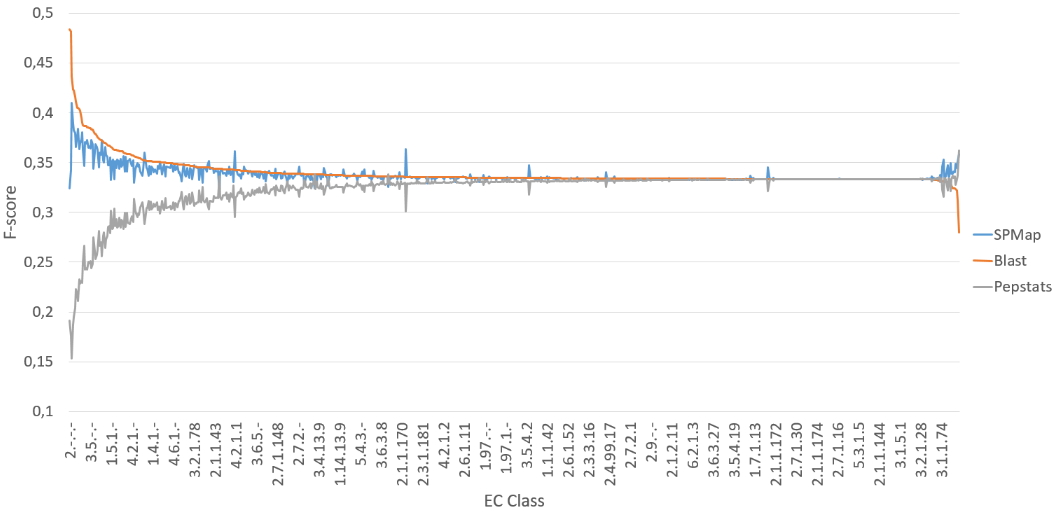


Figure 4.7: Weights of three independent EC classifiers based on BLAST-*k*NN classifier.

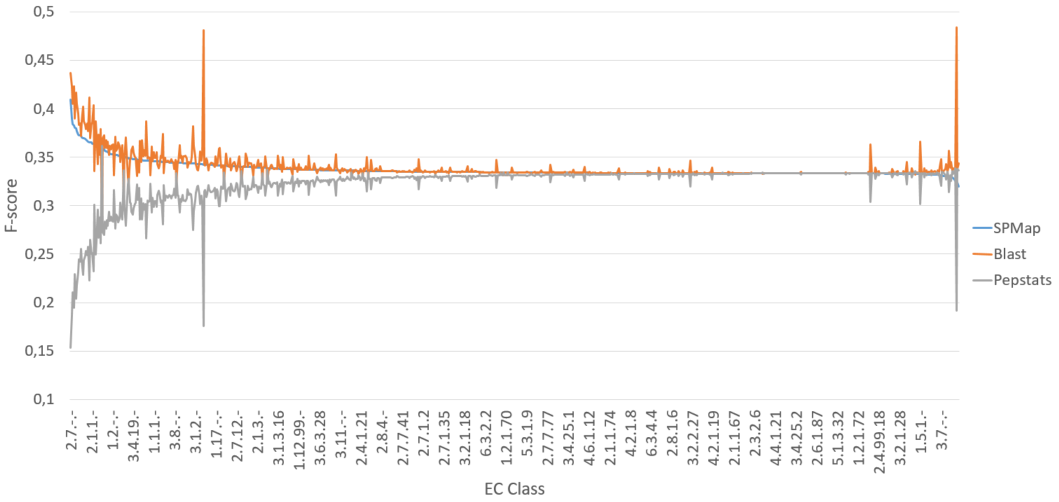


Figure 4.8: Weights of three independent EC classifiers based on SPMap classifier.

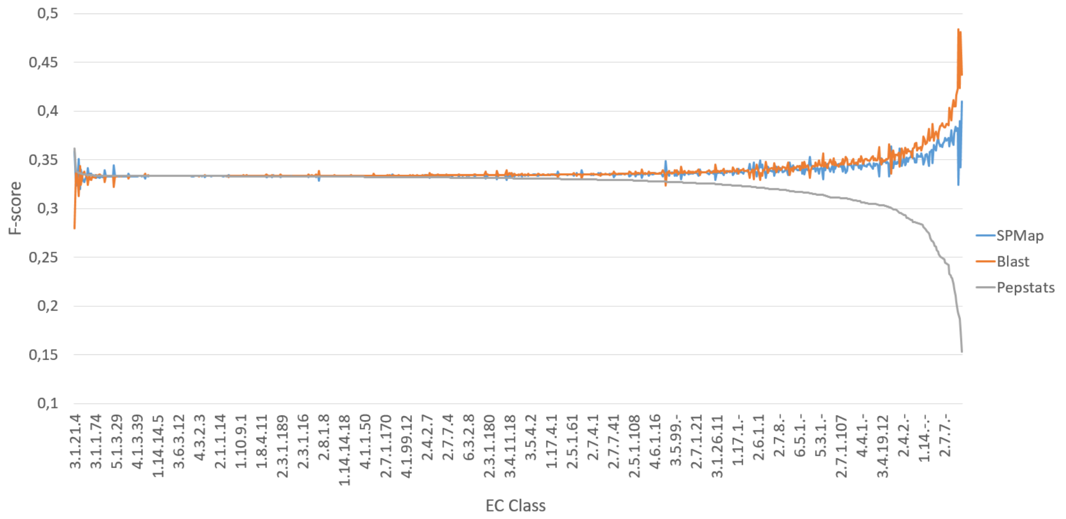


Figure 4.9: Weights of three independent EC classifiers based on Pepstats classifier.

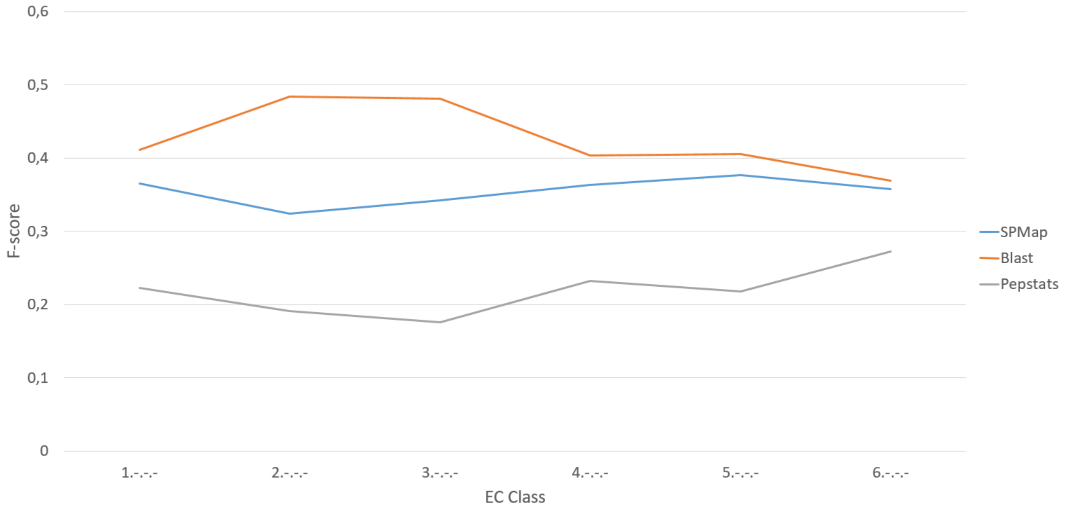


Figure 4.10: Weights of three independent EC classifiers for Level 1.

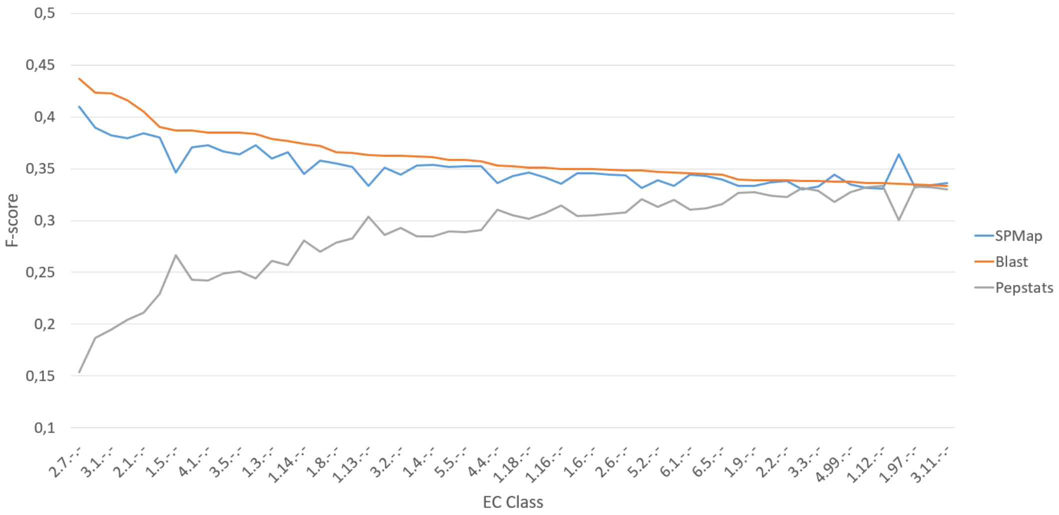


Figure 4.11: Weights of three independent EC classifiers for Level 2.

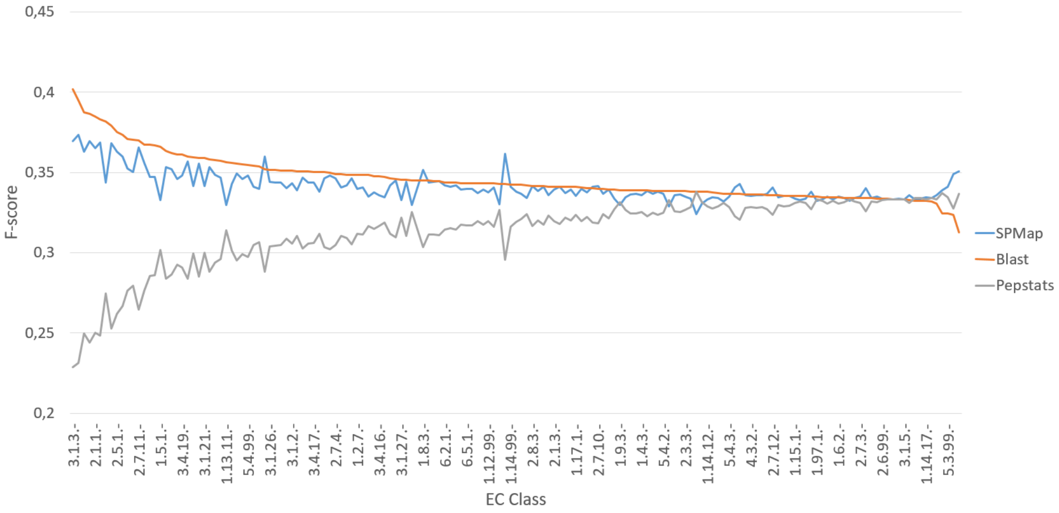


Figure 4.12: Weights of three independent EC classifiers for Level 3.

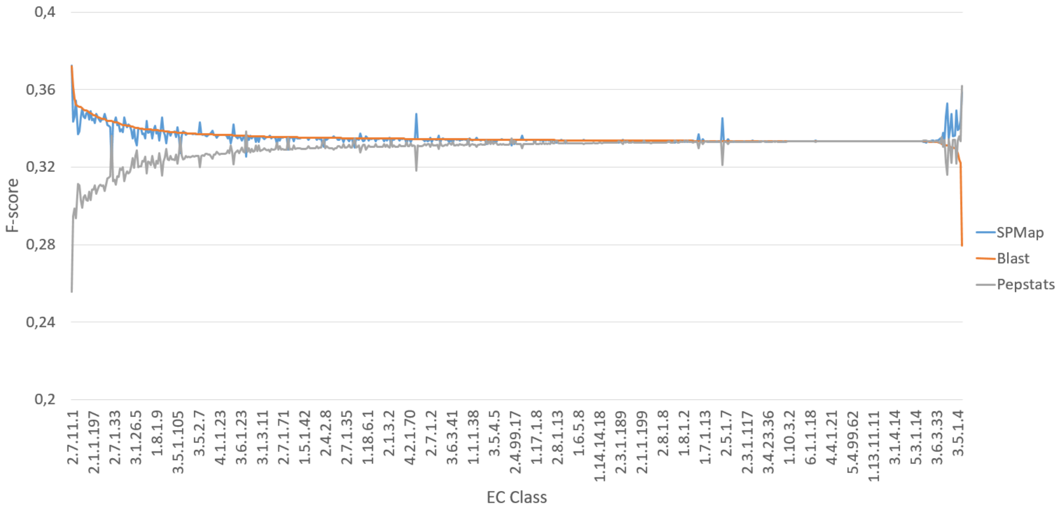
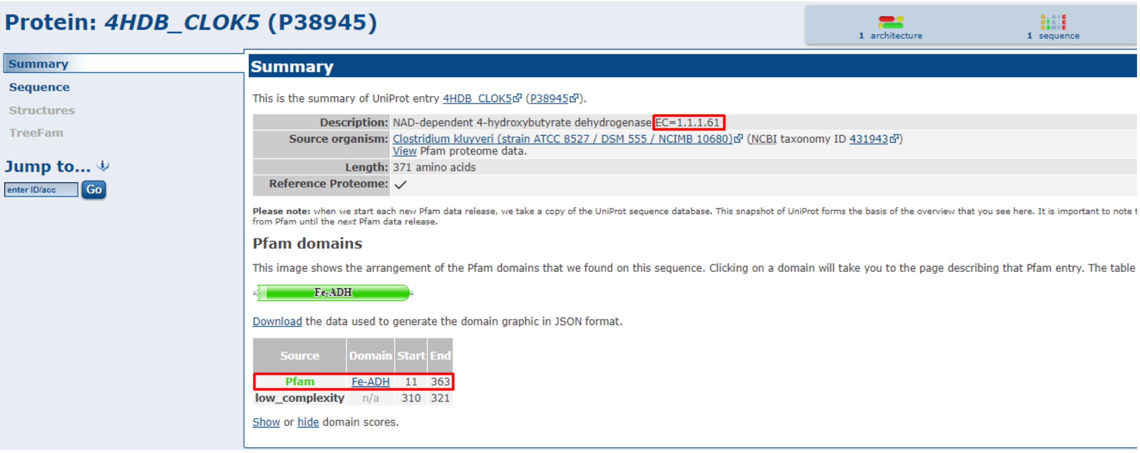


Figure 4.13: Weights of three independent EC classifiers for Level 4.

4.6 Predictions for Proteins That Have no Domain Annotation Information

Protein Families Database (Pfam) [38] uses functional domain information to assign EC number to a protein. Pfam results for P38945 are given in Figure 4.14. P38945 has a Pfam domain and annotated with EC number 1.1.1.61. Pfam fails to assign an EC number when the protein doesn't have a functional domain information. For example, Pfam result for protein E2JA32 is given in Figure 4.15. In UniProt, E2JA32 is annotated with EC number 6.3.2.46, however, it isn't annotated with any EC number in Pfam since it doesn't have Pfam domain information. To show that *ECPred* can predict proteins that don't have a functional domain information, a test set which consists of 50 enzyme proteins and 49 non-enzyme proteins which are not used in training datasets is constructed. For this set, 10 Oxidoreductases, 10 Transferases, 10 Hydrolases, 7 Lyases, 4 Isomerases and 9 Ligases are selected. Totally, 99 proteins are tested and average precision, recall and F-score values are calculated as 0.80, 0.48 and 0.60, respectively. Results show that, *ECPred* can predict non-enzymes with no Pfam domain information. *ECPred* can also predict main EC class of proteins, however, there is still room for improvement.



Protein: *4HDB_CLOK5* (P38945)

Summary

This is the summary of UniProt entry 4HDB_CLOK5 (P38945).

Description:	NAD-dependent 4-hydroxybutyrate dehydrogenase EC=1.1.1.61
Source organism:	Clostridium kluyveri (strain ATCC 8527 / DSM 555 / NCIMB 10680) (NCBI taxonomy ID 431943) View Pfam proteome data.
Length:	371 amino acids
Reference Proteome:	✓

Please note: when we start each new Pfam data release, we take a copy of the UniProt sequence database. This snapshot of UniProt forms the basis of the overview that you see here. It is important to note t
from Pfam until the *next* Pfam data release.

Pfam domains

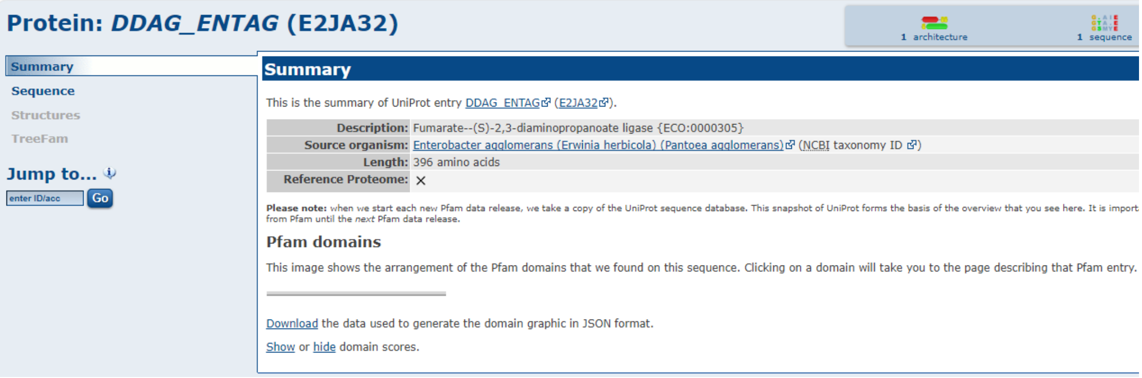
This image shows the arrangement of the Pfam domains that we found on this sequence. Clicking on a domain will take you to the page describing that Pfam entry. The table

Download the data used to generate the domain graphic in JSON format.

Source	Domain	Start	End
Pfam	Fe-ADH	11	363
low_complexity	n/a	310	321

Show or hide domain scores.

Figure 4.14: Pfam domain results for P38945 which has a function domain information and annotated with EC number 1.1.1.61.



Protein: *DDAG_ENTAG* (E2JA32)

Summary

This is the summary of UniProt entry DDAG_ENTAG (E2JA32).

Description:	Fumarate--(S)-2,3-diaminopropanoate ligase {ECO:0000305}
Source organism:	Enterobacter agglomerans (Erwinia herbicola) (Pantoea agglomerans) (NCBI taxonomy ID)
Length:	396 amino acids
Reference Proteome:	✕

Please note: when we start each new Pfam data release, we take a copy of the UniProt sequence database. This snapshot of UniProt forms the basis of the overview that you see here. It is import
from Pfam until the *next* Pfam data release.

Pfam domains

This image shows the arrangement of the Pfam domains that we found on this sequence. Clicking on a domain will take you to the page describing that Pfam entry.

Download the data used to generate the domain graphic in JSON format.

Show or hide domain scores.

Figure 4.15: Pfam domain results for E2JA32 which doesn't have a function domain information.

4.7 Comparisons with Other Tools

ECPred is compared with MecServer [24], ProtFun [5] and EzyPred [9] on enzyme/non-enzyme and main class tests. Two different test sets are used to compare methods. The first test set consists of 310 proteins that were initially non-enzyme and then, annotated with EC number in latest UniProt release and 720 non-enzyme human proteins that have an annotation score of 5 and which are never used in training. We also constructed the second test in order compare our results with EzyPred since their

Table 4.2: Enzyme or non-enzyme classification results of ProtFun and *ECPred* for the whole test set.

Method	**Precision**	**Recall**	**F-score**
ProtFun	0.39	**0.95**	0.56
ECPred	**0.84**	0.64	**0.72**

web-based tool accepts one input at a time and testing 1,030 proteins was not feasible one by one. Therefore, 30 proteins are selected among 310 positive proteins and 30 proteins are selected from 720 non-enzyme proteins for the second test.

4.7.1 Enzyme/non-enzyme comparison

Enzyme/non-enzyme comparison is made with ProtFun and EzyPred separately. MecServer results are not used in this set since MecServer only predicts enzyme main class.

4.7.1.1 Whole Set comparison with ProtFun

Totally, 1030 proteins are classified as enzyme/non-enzyme for this comparison. Whole set comparison results are given in Table 4.2. ProtFun is used for this comparison since the tool accepts multiple sequences as input and it is classified as enzyme or non-enzyme. ProtFun's recall value is higher than *ECPred* since ProtFun labels proteins mostly as enzyme, therefore the number of false positives increases and hence it has low precision value. On the contrary, the precision value is high in *ECPred* since we determined high threshold value in order to avoid FP. As a result, *ECPred* obtains significantly better F-score than ProtFun. Since training datasets of ProtFun is older than ECPred their performance is lower than ECPred.

4.7.1.2 Selected Proteins Comparison with EzyPred and ProtFun

30 enzyme proteins are selected from 310 positive test proteins and 30 non-enzyme proteins are selected from 720 negative test proteins for this comparison. These positive and negative proteins are selected in order to reflect the whole set performance of

Table 4.3: Enzyme or non-enzyme classification results of ProtFun, EzyPred and *ECPred* for selected proteins.

Method	Precision	Recall	F-score
ProtFun	0.72	**0.88**	**0.79**
EzyPred	0.37	0.37	0.37
ECPred	**0.91**	0.63	0.75

ECPred in this small test set. Performance results of these selected proteins for three methods are given in Table 4.3. Both ProtFun and *ECPred* outperforms EzyPred in this comparison. *ECPred* obtains significantly better results than ProtFun in the whole set. However, because of the selected proteins, ProtFun precision value is increased to 0.72, therefore, ProtFun's F-score is slightly better than that of *ECPred*.

4.7.2 Main Class Comparison

4.7.2.1 Whole Set Comparison with MecServer and ProtFun

The same 1030 proteins are also used for main class comparison. In this set, the flowchart which is given in Section 3.4 is followed. Firstly, a test protein is classified as enzyme or non-enzyme. Subsequently, its main class is determined if it is an enzyme. Whole set main class comparison results are given in Table 4.4. All three methods are obtained similar recall values which means they could not predict the proteins that annotated with an ECNumber in latest UniProt release. One important point; since MecServer does not give a prediction for enzyme/non-enzyme they are assigned whole non-enzyme proteins to one of the six main EC classes and therefore, they obtained low precision value. For this reason, their overall performance is low than *ECPred*. ProtFun is obtained similar performance results like MecServer. As a result, *ECPred* performance is better than ProtFun and MecServer in whole set main class comparison.

Table 4.4: Main class performance results of ProtFun, EzyPred and *ECPred* for the whole set.

Method	**Precision**	**Recall**	**F-score**
ProtFun	0.07	0.10	0.08
MecServer	0.07	0.16	0.09
ECPred	**0.51**	**0.13**	**0.40**

Table 4.5: Main class performance results for selected proteins. Results for proteins that *ECPred* obtained the highest performance in enzyme/non-enzyme prediction.

Method	**Precision**	**Recall**	**F-score**
ProtFun	0.10	0.20	0.14
EzyPred	0.09	0.20	0.12
MecServer	0.23	0.90	0.38
ECPred	**0.42**	**1.0**	**0.59**

4.7.2.2 Selected Proteins Comparison with MecServer, EzyPred and ProtFun

Selected 30 proteins are divided into three sets in order to make a fair comparison since all proteins are chosen subjectively. The first 10 positive test proteins are selected from 310 positive test proteins that *ECPred* is predicted correctly with the highest performance. 30 non-enzyme proteins are also added to this test set. Performance results of this test set are given in Table 4.5. Both ProtFun and EzyPred obtain very low precision and recall values and their F-score values are significantly lower than MecServer and *ECPred*. On highest performance test, *ECPred* gets slightly better recall values than MecServer and gets better precision results than MecServer as expected since MecServer does not predict non-enzymes. As a result, *ECPred* gets better results than ProtFun, EzyPred and MecServer.

The second 10 proteins are selected from 310 positive test proteins that *ECPred* gave the average performance and the same 30 non-enzyme proteins are used in this test set. Performance results of this test set are given in Table 4.6. ProtFun obtains the lowest performance results among four methods. EzyPred and MecServer obtain similar results which are slightly better than ProtFun. *ECPred* gets average recall values as expected since the selected proteins are chosen among average performance proteins. As a result, *ECPred* outperforms all three methods in average performance

Table 4.6: Results for proteins that *ECPred* obtained average performance in enzyme/non-enzyme prediction.

Method	Precision	Recall	F-score
ProtFun	0.06	0.10	0.07
EzyPred	0.09	0.20	0.12
MecServer	0.09	0.30	0.14
ECPred	**0.30**	**0.60**	**0.40**

Table 4.7: Results for proteins that *ECPred* obtained the lowest performance in enzyme/non-enzyme prediction.

Method	Precision	Recall	F-score
ProtFun	0.0	0.0	0.0
EzyPred	0.0	0.0	0.0
MecServer	**0.12**	**0.40**	**0.18**
ECPred	0.0	0.0	0.0

test set.

10 proteins are selected from 310 positive test proteins that *ECPred* gave the lowest performance and similarly, same 30 non-enzyme proteins are used in this test set. Performance results of this test set are given in Table 4.7. *ECPred*, ProtFun and EzyPred couldn't predict correctly any of the positive proteins, however, MecServer is predicted some of them correctly and got better results than other three methods. *ECPred* performance results are expected since the test proteins are selected from lowest performance proteins. EzyPred and ProtFun are also obtained the same results. However, MecServer made correct prediction some proteins that *ECPred* couldn't. MecServer got different results since they used different input feature set.

Finally, all these three sets are combined and overall performance is calculated. Overall performance results are given in Table 4.8. We chose these 60 proteins intentionally to obtain average results, therefore, it is expected to *ECPred* obtains average performance results. The important thing in here is the performance differences of other three methods from *ECPred*. ProtFun and EzyPred obtain the lowest performance results and MecServer obtains better performance results than these two methods. However, *ECPred* gets better results than all three methods. As a result, *ECPred* obtains the best performance results in main class comparison with selected proteins.

Table 4.8: Main class performance results for selected 60 proteins.

Method	Precision	Recall	F-score
ProtFun	0.15	0.10	0.12
EzyPred	0.16	0.13	0.15
MecServer	0.35	0.53	0.42
ECPred	**0.53**	**0.53**	**0.53**

4.8 Discussion

In this section, three proteins that are correctly predicted with *ECPred* are investigated. These proteins are annotated with EC numbers in latest UniProt release and not predicted correctly by ProtFun, MecServer and EzyPred. The first protein is Q96PB1 and annotated with EC number 2.3.1.45 which is defined as "N-acetylneuraminate 7-O (or 9-O)-acetyltransferase" at http://enzyme.expasy.org/EC/2.3.1.45. Q96PB1 is also known as human gene CASD1 manually annotated with the EC number 2.3.1.45 with literature curation in UniProt. Baumann et al. [39] conducted a lab experiment to prove that CASD1 is necessary for sialic acid 9-O-acetylation. The experiment results show that CASD1 is a sialate O-acetyltransferase and acts as a key enzyme in the biosynthesis of 9-O-acetylated sialoglycans .

The second protein is P31667 which is known as RPNA_ECOLI and annotated with EC number 3.1.21.- which is defined as "Endodeoxyribonucleases producing 5'-phosphomonoesters" at http://enzyme.expasy.org/EC/3.1.21.-. P31667 is also annotated with literature curation in UniProt. Kingston, Ponkratz and Raleigh [40] studied DNA-mobilizing enzymes; the recombination-promoting nuclease (RPN) families which belong to Pfam PF04754 and contains transposase_31 Pfam domain. Authors conducted a lab experiment and show that RPNA demonstrated magnesium-dependent, calcium- stimulated DNA endonuclease activity.

The third protein is A8LLX6 and annotated with EC number 4.2.1.- which is defined as "Hydro-lyases" at http://enzyme.expasy.org/EC/4.2.1.-. A8LLX6 is automatically annotated using the Unified Rule (UniRule): UR000031310. In this rule, a protein must meet three conditions. The first condition is a protein shouldn't contain fragment data. The second condition is a protein should belong to one of the Actinobacteria, Firmicutes and Proteobacteria taxonomies. The third and most important condition is a protein should match the High-quality Automated and Manual Annotation of Proteins (HAMAP) signature MF_01830. HAMAP is a system which uses manually curated family profiles to annotate proteins automatically. Since A8LLX6 meets all these conditions it is automatically annotated with EC number 4.2.1.-.

The main, subfamily and sub-subfamily EC classes of Q96PB1 is predicted correctly by *ECPred*, however, its substrate class is not predicted since we didn't train EC classes which are associated with less than 50 proteins. The main EC class of P31667 is predicted correctly by *ECPred*. The main and subfamily EC classes of A8LLX6 is predicted correctly by *ECPred*.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this study, EC hierarchy is denoted as follows; Level 0: enzyme or non-enzyme, Level 1: main class, Level 2: subfamily class, Level 3: sub-subfamily class and Level 4: substrate class. We developed a system called *ECPred*, which predicts automatically the enzymatic functions of proteins using a top-down approach. Firstly, the given sequence is classified as enzyme or non-enzyme. Subsequently, its subfamily class is determined. Sub-subfamily and substrate classes are also determined for a given sequence if the prediction scores for sub-subfamily and substrate classes are above the cut-off values. Combination of subsequence based, feature based and homology based methods are used to give a prediction score for query protein sequence. Totally, 858 EC classifiers are trained which consists of 6 main, 55 subfamily, 163 sub-subfamily and 634 substrate EC class classifiers.

Hierarchical data preparation is applied to each EC level and positive and negative training datasets are constructed separately for each EC class. Positive and negative optimal cut-off values are calculated for each EC class in order to give a positive and a negative prediction for a given input sequence.

In this thesis, we used an ensemble approach that combines three different approaches; homology based, subsequence based and feature based, respectively. This approach is previously applied to GO based functions [4] and it is shown to be effective, however, it is never applied to EC based enzymatic functions.

The average F-score value for validation dataset is calculated as 0.91, 0.98, 0.98 and 0.99 for EC main classes, subfamily classes, sub-subfamily classes and substrates classes, respectively. As a result, the average F-score value of 0.99 is obtained for all EC classes for the validation dataset. To the best of our knowledge, this is the first study that classifies 858 EC classes. According to the non-enzyme validation set results, *ECPred* obtained an average F-score of 0.98. However, six main classifiers predicted non-enzymes correctly at the same time with an F-score of 0.93 which is the true performance result of *ECPred* for the non-enzyme prediction.

Results of individual and combination of three independent classifiers are investigated. Results show that combination of three independent classifiers performance is higher than those of individual classifier for most of the EC classes. Additionally, weights of three independent classifiers are calculated. Results show that weights of BLAST*k*NN and SPMap are parallel to each other and weight of Pepstats-SVM is less than these two methods.

ECPred is compared with other similar tools on Level 0 and Level 1 tests: MecServer, ProtFun and EzyPred. Two independent test sets are constructed for these comparisons. Proteins that are annotated with an EC number in latest UniProt release are selected for the positive test set. Human non-enzyme proteins with an annotation score 5 from UniProt database is selected for the negative test set. Totally, 1030 test proteins are included in the first test set which consists of 310 enzymes as positives and 720 non-enzymes as negatives. For the second test, 30 proteins are selected from 310 test proteins that *ECPred* obtained average performance. 30 non-enzymes are also added to this test set. On the overall performance results, *ECPred* and MecServer get significantly higher F-score values than ProtFun and EzyPred and *ECPred* obtained slightly better performance than MecServer. As a result, performance results are not good as the validation set performance results, because, this type of rigorous test is never done in the past and usually, existing studies are measured their performances on their validation sets. However, *ECPred* obtained better results than existing tools on the independent test sets and the results show that there is still room for improvement.

Three proteins that are annotated with an EC number are investigated. Q96PB1 and P31667 are manually annotated using literature curation technique in UniProtKB. A8LLX6 is automatically annotated using UniRule: UR000031310. Their main EC classes are predicted correctly by *ECPred* since there are similar sequences in training datasets.

ECPred is used the protein sequence to obtain input features. There are other methods that use the structure information of the protein to predict enzymatic functions and their performance results are better than most of the existing studies. However, the structure information is not available for each protein. In addition, structure based methods are computationally intensive which means run times take too much time. Therefore, these type of methods are not practical for large scale analysis. Instead of structure based features, we used sequence based features since they are easy to obtain and fast to process.

In the future, we plan to develop a web-based tool and a standalone tool which takes a protein sequence and predicts whether the given protein is an enzyme or a non-enzyme, and the exact enzymatic function if it is predicted as an enzyme. The tool will follow the flowchart which is given Section 3.4, therefore, not all of the 858 EC classifiers will be running for a query protein sequence. Since we only trained 858 EC classes we are planning to add more EC classes to *ECPred* in future. New input feature can be added to the system.

REFERENCES

[1] E. C. Webb, K. P., and L. K.L., *Enzyme nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes.* Academic Press, 1992, no. Ed. 6.

[2] A. Yaman, "Prediction of Enzyme Classes in a Hirarchical Approach by Using SPMap," Master's thesis, METU, Turkey, 2009.

[3] A. S. Rifaioglu, "An Extension to GOPred to Annotate Swiss-Prot and TREMBL Sequences for all Gene Ontology Categories and EC Numbers," Master's thesis, METU, Turkey, 2015.

[4] Ö. S. Saraç, V. Atalay, and R. Cetin-Atalay, "Gopred: Go molecular function prediction by combined classifiers," *PloS one*, vol. 5, no. 8, p. e12382, 2010.

[5] L. J. Jensen, R. Gupta, N. Blom, D. Devos, J. Tamames, C. Kesmir, H. Nielsen, H. H. Stærfeldt, K. Rapacki, C. Workman *et al.*, "Prediction of human protein function from post-translational modifications and localization features," *Journal of molecular biology*, vol. 319, no. 5, pp. 1257–1265, 2002.

[6] P. D. Dobson and A. J. Doig, "Distinguishing enzyme structures from non-enzymes without alignments," *Journal of molecular biology*, vol. 330, no. 4, pp. 771–783, 2003.

[7] ——, "Predicting enzyme class from protein structure without alignments," *Journal of molecular biology*, vol. 345, no. 1, pp. 187–199, 2005.

[8] L. C. Borro, S. R. Oliveira, M. E. Yamagishi, A. L. Mancini, J. G. Jardine, I. Mazoni, E. D. Santos, R. H. Higa, P. R. Kuser, and G. Neshich, "Predicting enzyme class from protein structure using bayesian classification," *Genet. Mol. Res*, vol. 5, no. 1, pp. 193–202, 2006.

[9] H.-B. Shen and K.-C. Chou, "Ezypred: a top–down approach for predicting enzyme functional classes and subclasses," *Biochemical and biophysical research communications*, vol. 364, no. 1, pp. 53–59, 2007.

[10] X.-B. Zhou, C. Chen, Z.-C. Li, and X.-Y. Zou, "Using chou's amphiphilic pseudo-amino acid composition and support vector machine for prediction of enzyme subfamily classes," *Journal of theoretical biology*, vol. 248, no. 3, pp. 546–551, 2007.

[11] W.-L. Huang, H.-M. Chen, S.-F. Hwang, and S.-Y. Ho, "Accurate prediction of enzyme subfamily class using an adaptive fuzzy k-nearest neighbor method," *Biosystems*, vol. 90, no. 2, pp. 405–413, 2007.

[12] L. Lu, Z. Qian, Y.-D. Cai, and Y. Li, "Ecs: an automatic enzyme classifier based on functional domain composition," *Computational biology and chemistry*, vol. 31, no. 3, pp. 226–232, 2007.

[13] E. Nasibov and C. Kandemir-Cavas, "Efficiency analysis of knn and minimum distance-based classifiers in enzyme family prediction," *Computational biology and chemistry*, vol. 33, no. 6, pp. 461–464, 2009.

[14] J.-D. Qiu, S.-H. Luo, J.-H. Huang, and R.-P. Liang, "Using support vector machines to distinguish enzymes: approached by incorporating wavelet transform," *Journal of theoretical biology*, vol. 256, no. 4, pp. 625–631, 2009.

[15] D. A. Latino and J. Aires-de Sousa, "Assignment of ec numbers to enzymatic reactions with molmap reaction descriptors and random forests," *Journal of chemical information and modeling*, vol. 49, no. 7, pp. 1839–1846, 2009.

[16] Y.-C. Wang, X.-B. Wang, Z.-X. Yang, and N.-Y. Deng, "Prediction of enzyme subfamily class via pseudo amino acid composition by incorporating the conjoint triad feature," *Protein and Peptide Letters*, vol. 17, no. 11, pp. 1441–1449, 2010.

[17] N. J. Davidson and X. Wang, "Non-alignment features based enzyme/non-enzyme classification using an ensemble method," in *Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on*. IEEE, 2010, pp. 546–551.

[18] Y.-C. Wang, Y. Wang, Z.-X. Yang, and N.-Y. Deng, "Support vector machine prediction of enzyme function with conjoint triad feature and hierarchical context," *BMC systems biology*, vol. 5, no. 1, p. S6, 2011.

[19] L. De Ferrari, S. Aitken, J. van Hemert, and I. Goryanin, "Enzml: multi-label prediction of enzyme classes using interpro signatures," *BMC bioinformatics*, vol. 13, no. 1, p. 61, 2012.

[20] C. Kumar and A. Choudhary, "A top-down approach to classify enzyme functional classes and sub-classes using random forest," *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2012, no. 1, p. 1, 2012.

[21] V. Volpato, A. Adelfio, and G. Pollastri, "Accurate prediction of protein enzymatic class by n-to-1 neural networks," *BMC bioinformatics*, vol. 14, no. 1, p. S11, 2013.

[22] Y. Matsuta, M. Ito, and Y. Tohsato, "Ecoh: an enzyme commission number predictor using mutual information and a support vector machine," *Bioinformatics*, vol. 29, no. 3, pp. 365–372, 2012.

[23] C. Nagao, N. Nagano, and K. Mizuguchi, "Prediction of detailed enzyme functions and identification of specificity determining residues by random forests," *PloS one*, vol. 9, no. 1, p. e84623, 2014.

[24] Y. Che, Y. Ju, P. Xuan, R. Long, and F. Xing, "Identification of multi-functional enzyme with multi-label classifier," *PloS one*, vol. 11, no. 4, p. e0153503, 2016.

[25] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[26] K.-C. Chou, "Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes," *Bioinformatics*, vol. 21, no. 1, pp. 10–19, 2004.

[27] K.-C. Chou and D. W. Elrod, "Prediction of enzyme family classes," *Journal of Proteome Research*, vol. 2, no. 2, pp. 183–190, 2003.

[28] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor, "Kernel-based learning of hierarchical multilabel classification models," *Journal of Machine Learning Research*, vol. 7, no. Jul, pp. 1601–1626, 2006.

[29] P. Rice, I. Longden, and A. Bleasby, *EMBOSS: the European molecular biology open software suite*. Elsevier Current Trends, 2000.

[30] E. Gasteiger, C. Hoogland, A. Gattiker, S. Duvaud, M. R. Wilkins, R. D. Appel, and A. Bairoch, *Protein identification and analysis tools on the ExPASy server*. Springer, 2005.

[31] O. S. Sarac, Ö. Gürsoy-Yüzügüllü, R. Cetin-Atalay, and V. Atalay, "Subsequence-based feature map for protein function classification," *Computational biology and chemistry*, vol. 32, no. 2, pp. 122–130, 2008.

[32] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, and U. Consortium, "Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches," *Bioinformatics*, vol. 31, no. 6, pp. 926–932, 2014.

[33] L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li, "Cd-hit: accelerated for clustering the next-generation sequencing data," *Bioinformatics*, vol. 28, no. 23, pp. 3150–3152, 2012.

[34] T. Madden, "The blast sequence analysis tool," 2013.

[35] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[36] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks," *Proceedings of the National Academy of Sciences*, vol. 89, no. 22, pp. 10 915–10 919, 1992.

[37] T. Joachims, “Making large-scale svm learning practical,” Technical Report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund, Tech. Rep., 1998.

[38] R. D. Finn, P. Coggill, R. Y. Eberhardt, S. R. Eddy, J. Mistry, A. L. Mitchell, S. C. Potter, M. Punta, M. Qureshi, A. Sangrador-Vegas *et al.*, “The pfam protein families database: towards a more sustainable future,” *Nucleic acids research*, vol. 44, no. D1, pp. D279–D285, 2016.

[39] A.-M. T. Baumann, M. J. Bakkers, F. F. Buettner, M. Hartmann, M. Grove, M. A. Langereis, R. J. De Groot, and M. Mühlenhoff, “9-o-acetylation of sialic acids is catalysed by casd1 via a covalent acetyl-enzyme intermediate,” *Nature communications*, vol. 6, 2015.

[40] A. W. Kingston, C. Ponkratz, and E. A. Raleigh, “Rpn (yhga-like) proteins of escherichia coli k-12 and their contribution to reca-independent horizontal transfer,” *Journal of bacteriology*, vol. 199, no. 7, pp. e00 787–16, 2017.