Alperen Emre Ulutaş

PROTEIN FRAGMENT SELECTION USING MACHINE LEARNING

AGU

2018

PROTEIN FRAGMENT SELECTION USING MACHINE LEARNING

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF NATURAL SCIENCES OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER

By
Alperen Emre Ulutaş
May 2018

PROTEIN FRAGMENT SELECTION USING MACHINE LEARNING

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF NATURAL SCIENCES OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER

By
Alperen Emre Ulutaş
May 2018

SCIENTIFIC ETHICS COMPLIANCE

I hereby declare that all information in this document has been obtained 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 materials and results that are not original to this work.

Alperen Emre Ulutaş

REGULATORY COMPLIANCE

M.Sc. thesis titled **“PROTEIN FRAGMENT SELECTION USING MACHINE LEARNING”** has been prepared in accordance with the Thesis Writing Guidelines of the Abdullah Gül University, Graduate School of Engineering & Science.

Prepared By	Advisor
Alperen Emre Ulutaş	Dr. Zafer Aydın

Head of the Electrical and Computer Engineering Program

Assoc. Prof. Vehbi Çağrı GÜNGÖR

ACCEPTANCE AND APPROVAL

M.Sc. thesis titled Protein Fragment Selection Using Machine Learning and prepared by Alperen Emre Ulutaş has been accepted by the jury in the Electrical and Computer Engineering Graduate Program at Abdullah Gül University, Graduate School of Engineering & Science.

……….. /……….. / ………..

(Thesis Defense Exam Date)

JURY:

Dr. Zafer Aydın :…………………….

Dr. Bekir Hakan Aksebzeci :……………………..

Doç.Dr.Celal Öztürk :……………………..

APPROVAL:

The acceptance of this M.Sc. thesis has been approved by the decision of the Abdullah Gül University, Graduate School of Engineering & Science, Executive Board dated ….. /….. / ……….. and numbered ………….……… .

……….. /……….. / ………..

(Date)

Graduate School Dean

Prof. Irfan ALAN

ABSTRACT

PROTEIN FRAGMENT SELECTION USING MACHINE LEARNING

Alperen Emre ULUTAŞ

M.Sc. in Electrical and Computer Engineering Department

Supervisor: Dr. Zafer Aydın

May-2018

Protein fragment selection is an important step in predicting the three-dimensional (3D) structure of proteins. Selecting the right fragments contributes significantly to accurate prediction of 3D structure. In this thesis, a machine learning approach is employed to predict whether a pair of protein fragments have similar 3D structures or not, which can be used to select fragment structures for a target protein with unknown structure. To design input features, a concepy hierarchy is implemented, which considers sequence profile matrices, predicted secondary structure, solvent accessibility and torsion angle classes as features in various combinations and projections. Several machine learning classifiers and regressors are trained and optimized for predicting the structural similarity of 3-mer and 9-mer fragments including logistic regression, AdaBoost, decision tree, k-nearest neighbor, naive Bayes, random forest, SVM and multi-layer perceptron. The results demonstrate that combining different feature sets through concept hierarcy and model optimization improves the prediction accuracy substantially. Furthermore it is possible to predict the structural similarity of fragment pairs with high accuracy, which is assessed by perforing cross-validation experiments on fragment datasets. When the structural similarity of fragments is defined as a classification problem, the accuracy of different classifiers are obtained as similar to each other. Among the regression methods, random forest provided the best accuracy metrics.

Keywords: Protein Fragment Selection, Protein Structure Prediction, Secondary Structure Prediction, Solvent Accessibility Prediction, Torsion Angle Prediction

ÖZET

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE PROTEİN PARÇACIK SEÇİMİ

Alperen Emre ULUTAŞ

Elektrik ve Bilgisayar Mühendisliği Bölümü Yüksek Lisans

Tez Yöneticisi: Dr. Zafer Aydın

Mayıs-2018

Protein parçacık seçimi proteinlerin üç boyutlu yapılarının tahmin edilmesindeki önemli adımlardan biridir. Doğru parçacık yapılarının seçilmesi üç boyutlu yapının doğru tahmin edilmesi için gereklidir. Bu tezde verilen iki protein parçacığının üç boyutlu yapılarının birbirine benzer olup olmadığını tahmin eden çeşitli yapay öğrenme yöntemleri geliştirilmiştir. Bu sayede yapısı bilinmeyen bir hedef protein için parçacık yapılarının seçilmesi mümkün olacaktır. Tahmin yönteminin girdi olarak kullanacağı öznitelik parametrelerinin tasarlanması için bir konsept hiyerarşi yaklaşımı izlenmiştir. Bunun için dizi profil matrisleri, ikincil yapı, çözücü erişilirlik ve bükülme açı sınıfı tahminleri çeşitli kombinasyonlarda ve izdüşüm uzaylarında incelenmiştir. Üç ve dokuz amino asitlik parçacıkların yapısal benzerlik tahmini için çeşitli sınıflandırma ve regresyon modelleri eğitilmiş ve optimize edilmiştir. Bunlar arasında lojistik regresyon, AdaBoost, karar ağacı, en yakın komşu, sade Bayes, rastgele orman, destek vektör makinası ve çok-katmanlı algılayıcı bulunmaktadır. Elde edilen sonuçlara göre farklı öznitelik kümelerinin konsept hiyerarşi yaklaşımı ile birleştirilmesi ve model optimizasyonları tahmin başarısını önemli oranda iyileştirmiştir. Ayrıca çapraz doğrulama deneyleri neticesinde parçacık benzerliğinin yüksek başarı oranları ile tahmin edilebildiği gösterilmiştir. Parçacık benzerliği sınıflandırma problemi olarak tanımlandığı zaman tahmin yöntemlerinin başarı oranları birbirine yakın olarak elde edilmiştir. Regresyon modelleri arasında ise rastgele orman yöntemi en yüksek tahmin başarısına ulaşmıştır.

Keywords: Protein Parçacık Seçimi, Protein Yapı Tahmini, İkincil Yapı Tahmini, Çözücü Erişilirlik Tahmini, Bükülme Açısı Tahmini

Acknowledgements

I would like to thank my advisor Dr. Zafer AYDIN for his goodwill and help on my study.

I would like to thank my familiy members my father Zafer ULUTAŞ, my mother Nuray ULUTAŞ and my sister ESRA ULUTAŞ for their support.

The results reported here were partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).

This work is supported by grant 113E550 from 3501 TUBITAK National Young Researchers Career Award.

Table of Contents

List of Figures

List of Tables

This thesis is dedicated to my family

Chapter 1

Introduction

Proteins are large organic molecules that contain amino acids as building blocks. Each protein has a unique amino acid sequence specified by the genetic code in DNA. Proteins play important roles in biological processes such as metabolic reactions, pathway regulation, growth, formation of skin, hair, blood cells, muscles and bones.

The biological function of a protein is closely related to its structure. There are two major approaches for solving the structure of a protein: experimental and computational. Experimental techniques include X-ray diffraction and nuclear magnetic resonance (NMR), which can be expensive, time consuming and may not even be applied to certain proteins. As an alternative, computational prediction of protein structure can be perfromed for any protein, which is cheaper and more efficient though typically less accurate than experimental methods. In addition to revealing the functional role of proteins, solving protein structure either experimentally or computationally can also be used to model protein-ligand interactions in drug and enzyme design. Therefore, accurate determination of protein structure contributes positively to the success of function prediction and drug design.

Computational prediction of three-dimensional structure is a challenging problem. Instead of searching the optimum structure conformation directly, which might be computationally costly, first, various one or two-dimensional properties of the target protein are computed such as sequence profiles, predictions of secondary structure, solvent accessibility, torsion angles, contact maps and selection of fragments, which are used as inputs to subsequent steps in the pipeline. This work contentrates on the fragment selection problem.

The following sections include a brief overview of protein structure, literature review and contributions of this thesis.

1.1 Protein Structure

Proteins are formed by amino acids that are bound together in a consecutive manner with peptide bonds and are important macromolecules for every organism. There are 20 amino acids that are commonly found in nature. An amino acid is an organic composite that contains an amine group (-NH2), a carboxyl group (-COOH) and a side chain molecule (R) bounding to an asymmetric alpha carbon atom (Ca). An example amino acid structure is given below in Figure 1.1.1. Each amino acid has different pyhsical and chemical properities such as electrostatic charge, hydrophobicity condition, acid decomposition coefficient (pKa), size and functional groups, which play an important role in determining the structure of a protein [1].

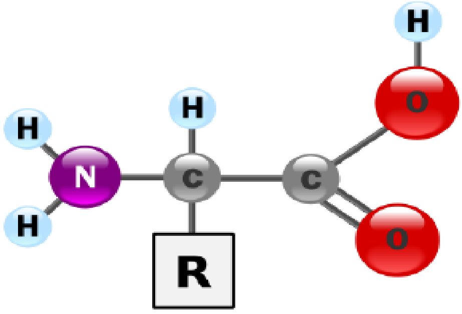


Figure 1.1.1 Structure of an amino acid [2]

1.1.1 Protein Structure Levels

Protein structure has four fundamental levels: primary, secondary, tertiary and quaternary. Primary structure is the amino acid sequence; secondary structure is formed by repetitive the hydrogen bonding patterns; tertiary structure refers to the three-dimensional structure of a single amino acid chain; and quaternary structure is the three dimensional structure of multiple amino acid chains.

1.1.1.1 Primary Structure

The peptide bonds that from during protein synthesis makes the primary structure stay together. In living being's the gene that codes the protein is what determines the primary structure. The amino acid sequence is unique for that protein and determines its 3D structure and function. The sequence of amino acids that make a protein can be extracted by translating the gene's sequence that has coded the protein [1]. There are also other methods that can be used to find the amino acid content of proteins such as Edman degradation and mass spectrometry (MS). The primary structure starts from the amino acid at the N-terminal, which has a flanking amide group and ends at the amino acid at the C-terminal, which has a flanking carboxyl group. Figure 1.1.1.1.1 shows the primary structure of a protein.

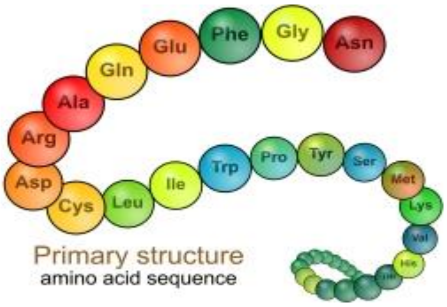


Figure 1.1.1.1.1 Primary structure of a protein [3]

1.1.1.2 Secondary Structure

Secondary structure contains regular hydrogen bonds formed between neighbouring amino acids that have similar torsion angles. Such amino acids come together to form secondary structure segments. There are two types of hydrogen bonding patterns: the rotation motif and the bridge motif. In rotation motif, also called the n-rotation motif, there is a hydrogen bond between the amino acid at position *i* and the amino acid at position *i+n*, where *n* typically takes values equal to 3, 4 or 5. In bridge motif, there are hydrogen bonds between amino acids that may not be close to each other with regard to sequence. Secondary structure is formed when the rotation and bridge motifs come together in a consecutive and specific manner. For example, rotation motif that repeats 4 times forms the alpha helices and repeating bridge motifs form beta sheets, which may contain multiple beta-strand segments. On the other hand loops typically contain irregular bonding patterns. The tertiary structure of the protein can be tought as the secondary structure elements assembled together. Figure 1.1.1.2.1, shows the secondary structure of a protein.

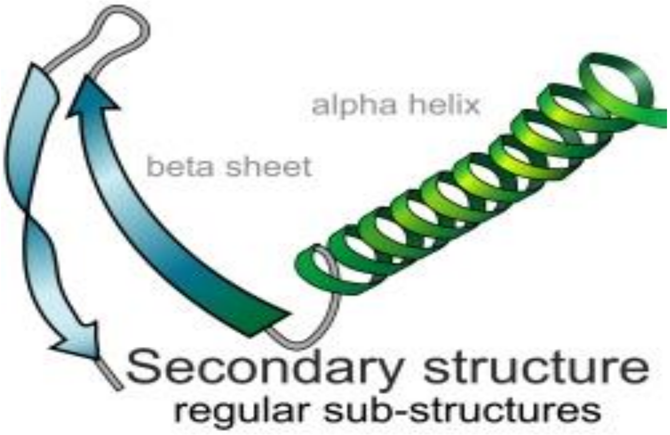


Figure 1.1.1.2.1 Secondary structure of a protein [4]

1.1.1.2.1 Helix

In helices, protein's backbone forms a spiral structure (Figure 1.1.1.2.1.1). There are three types of helices: alpha helix (α-helix), 3_{10} helix and the pi helix (π-helix). Helices can have various functional roles. Motifs that bind the DNA (strand-coil-strand, leucine zipper, zinc finger) and structures that go through the cell membrane (rhodopsins, G-protein clamped receptors) are among the examples to helical structures [5].

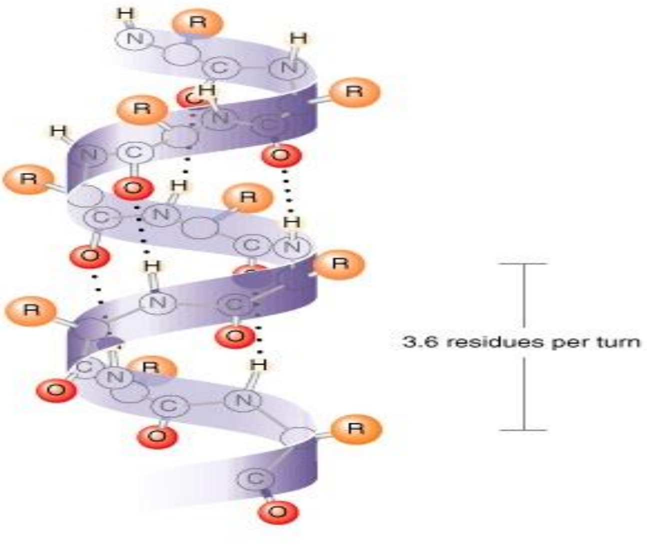


Figure 1.1.1.2.1.1 Alpha helix [6]

1.1.1.2.2 Beta Strand and Beta Sheet

Beta sheets are formed by beta strands that interact pairwise through hydrogen bonds (Figure 1.1.1.2.2.1). A beta sheet should contain at least two beta strands and each beta strand needs at least 2 or 3 hydrogen bonds that make connections to its partner strand. A beta strand segment typically has a length between 3 to 10 amino acids. Interacting amino acids in beta-strand segments can be either close to each other or far from each other according to the amino acid sequence. Those beta-strands that are far apart based on the one-dimensional sequence may come closer when the protein molecule folds into its 3D structure. Protein aggregation and fibrills which form due to merging of beta sheets have roles in various diseases such as Alzheimer's [7].

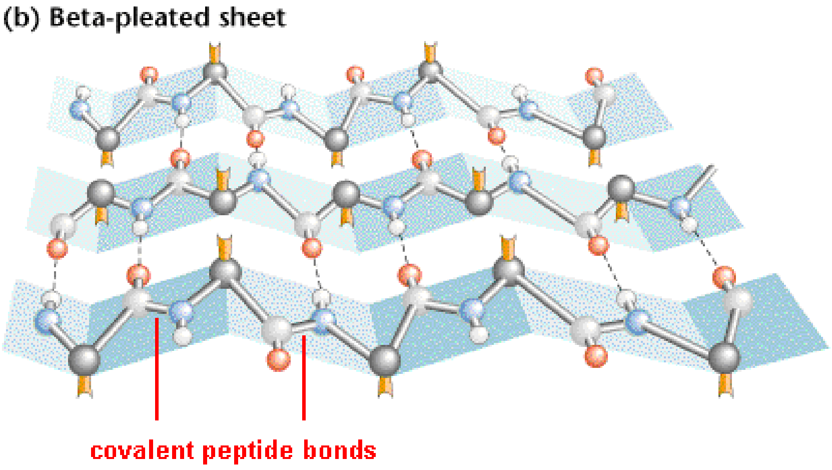


Figure 1.1.1.2.2.1 Beta Sheet [8]

1.1.1.2.3 Loop

Loops are structures that are mostly present between helices and beta sheets, with different lengths and confugrations. They are usually located on the surface area of proteins. Loops do not impose strong constraints in secondary structure aligments because there can be more mutations (substitution and deletion) in loop structures than in helices or beta-strands. Loops tend to have charged and polarized amino acids and are usually found in functionally more active regions [9]. There are 3 types of loops: turn, bend and random coil.

1.1.1.3 Tertiary Structure

Tertiary structure is the three-dimensional (3D) coordinates of the atoms in a protein. The folding of protein is guided by chaperon proteins and hydrophobic interactions. Furthermore for the structure to be stable, specific tertiary interactions (e.g. salt bridges, hydrogen bonds, disilfude bonds and stacking of side bonds) may be needed [1]. An example to tertiary structure is given below in Figure 1.1.1.3.1.

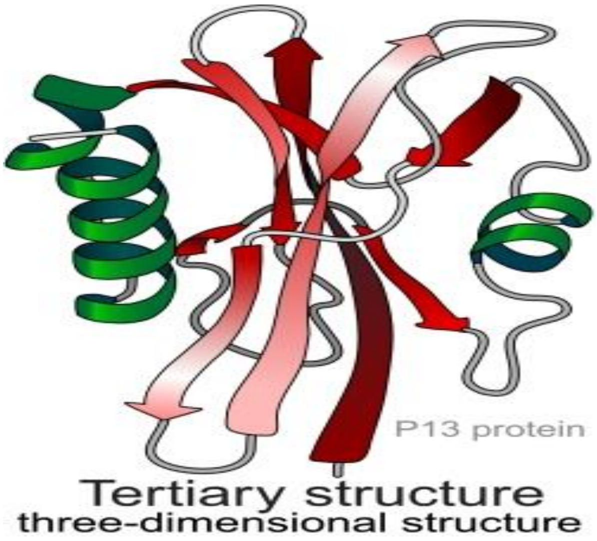


Figure 1.1.1.3.1 Protein tertiary structure [10]

1.1.1.3.1 Dihedral Angles

Dihedral (torsion) angles are the rotation angles of specific bonds on the protein backbone. There are three types of dihedral angles. Omega (ω) is the angle of rotation around the peptide bond, phi (ϕ) is the angle of rotation between N and the C_α atom, psi (ψ) is the the angle of rotation between between the C=O and the C_α atom (Figure 1.1.1.3.1.1). The omega angle does not show flexibility and usually takes values close to 180 degrees. Phi and psi angles on the other hand, can take different values. These are the internal liberty angles of a protein and control protein conformation. In some secondary structure elements the values these angles can receive are limited due to geometric concerns (Ramachandran graphic) [1].

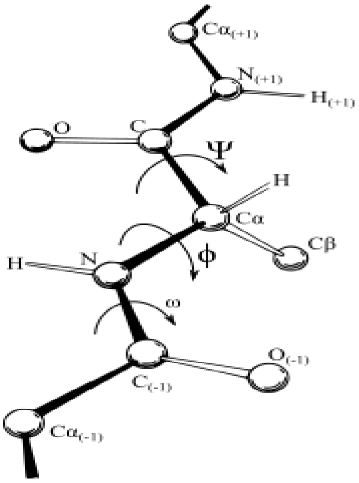


Figure 1.1.1.3.1.1 Dihedral angles [11]

1.1.1.3.2 Solvent Accesibility

Solvent accessibility is the accesible surface area of a biomolecule that can be reached by a solvent such as water. Van der Waals area is proporional to atoms' diameters and can be defined as the surface of the red circles shown in Figure 1.1.1.3.2.1 as dashed lines. According to this figure, the accesible surface can obtained by tracking a representative solvent molecule (blue circle) on the Waals surface. Amino acids that are in inner parts of the protein are less accesible to a solvent compared to the amino acids that are closer to the surface.

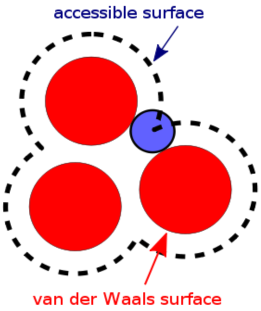


Figure 1.1.1.3.2.1 Solvent accesibility [12]

1.1.1.4 Quaternary Structure

Quaternary structure is formed by several proteins or polypeptide chains gathered together (Figure 1.1.1.4.1). It may contain non-covalent and disulfide bonds, which stabilize the overal 3D structure. Most proteins do not have a quaternary structure and function as monomeric units.

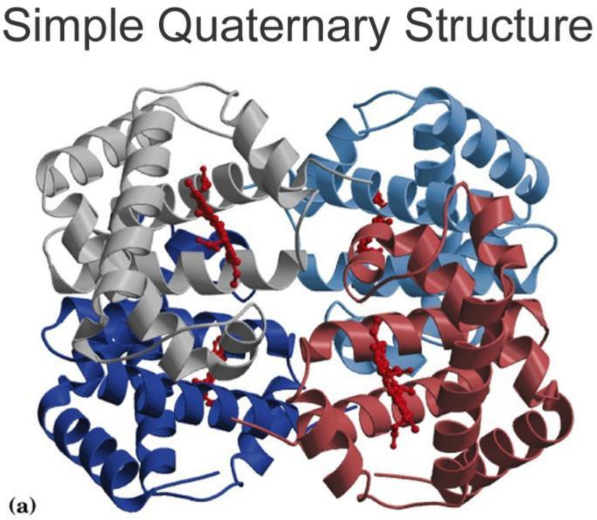


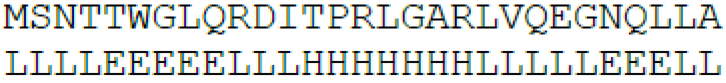
Figure 1.1.1.4.1 Protein quaternary structure [13]

1.2 Protein Structure Prediction

To date, there has been many studies on protein structure prediction. Nevertheless, the problem hasn’t been solved completly yet. Due to the challenges in searching for the best 3D structure directly, the problem is divided into sub-parts. First the target protein is compared with proteins in the database by various alignment algorithms, which can be used to compute statistical profile matrices based on the frequency of occurrence counts of the amino acids in specific positions. These matrices can be employed as input features to predict certain properties of protein structure such as secondary structure, dihedral angles, solvent accesibility, disordered regions, contact maps [14]. In the next step fragment structures are selected for overlapping segments of the target protein. These predictions and fragments provide constraints for and reduce the search space of the 3D structure prediction algorithms considerably.

1.2.1 Secondary Structure Prediction

Protein secondary structure prediction aims to assign a secondary structure class label to each amino acid of a protein (Figure 1.2.1.1). In 3-state prediction, the classes are helix (H), beta-strand (E) and loop (L). Typically supervised learning methods are used for secondary structure prediction. For this purpose, proteins whose secondary structure is known in the protein database are used to train a model, which predicts the secondary structure of a protein. Preliminary methods for secondary structure depended on the tendency of each amino acids to prefer one type of secondary structure over the others. In addition, free energy rules for the formation of secondary structure elements were included. These methods were 60% succesfull in predicting the state the amino acid will adopt (helix, beta-strand, or loop). When multiple sequence alignments are used as input features, there has been an important increase in the accuracy of predictions to 80-82% [15-16]. Furthermore, using template proteins whose structure is known and similar to the target protein the accuracy increased to around 84-85% [17-18]. These results made it possible to use the information from secondary structure prediction on other problems such as folding class prediction, three-dimensional (3D) structure prediction, classifying structural motifs and improving sequence alignments.

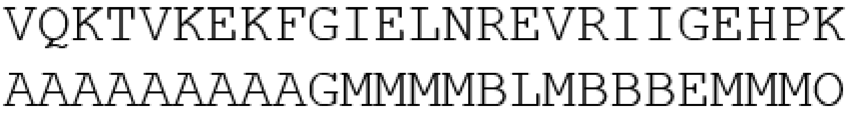


```
MSNTTWGLQRDITPRLGARLVQEGNQLLA
LLLLEEEEELLLHHHHHHHLLLLLEEELL
```

Figure 1.2.1.1 Secondary structure prediction. First line is the amino acid sequence, second line is the secondary structure class labels (H:Helix, E:Beta Strand, L:Loop)

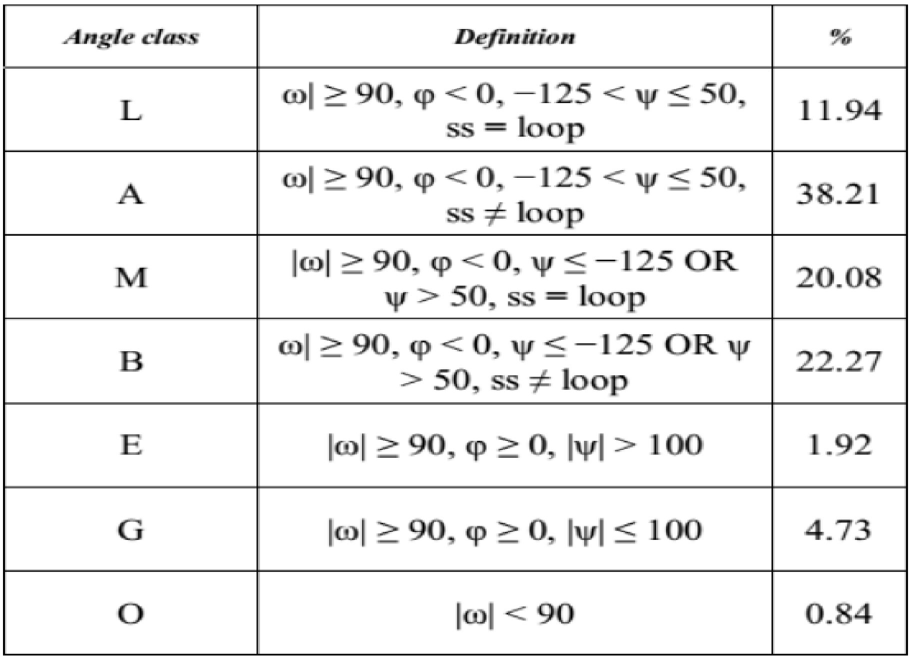
1.2.2 Dihedral Angle Prediction

As shown in Table 1.2.2.1, for each amino acid there are three types of dihedral angles: phi (φ), psi (ψ) and omega (ω). The aim of dihedral angle prediction is to predict these continuous valued angles for each amino acid of a protein. Since, omega (ω) angle usually takes values close to 180 degrees, sometimes the problem can be defined as predicting the phi (φ) and psi (ψ) angles only, which form groups for various secondary structure elements on the Ramachandran plot. For this reason, in another version of the dihedral angle prediction problem, phi (φ), psi (ψ) and omega (ω) angles are mapped to discrete classes in which case the torsion angle classes are estimated. This approach is advantageous for machine learning techniques because it is easier to predict discrete classes than to predict continuous valued angles. Furthermore for three-dimensional (3D) structure prediction, torsion angle class information could still be useful. Figure 1.2.2.1 summarizes the seven state dihedral angle class prediction problem, which aims to assign a dihedral angle class to each amino acid from a seven letter alphabet.



```
VQKTVKEKFGIELNREVRIIGEHPK
AAAAAAAAAGMMMMBLMBBBEMMMO
```

Figure 1.2.2.1 Seven state dihedral angle prediction. First line is the amino acid sequence, second line is the dihedral angle class labels.



Angle class	*Definition*	*%*
L	$\omega| \geq 90, \varphi < 0, -125 < \psi \leq 50$, ss = loop	11.94
A	$\omega| \geq 90, \varphi < 0, -125 < \psi \leq 50$, ss ≠ loop	38.21
M	$|\omega| \geq 90, \varphi < 0, \psi \leq -125$ OR $\psi > 50$, ss = loop	20.08
B	$\omega| \geq 90, \varphi < 0, \psi \leq -125$ OR $\psi > 50$, ss ≠ loop	22.27
E	$|\omega| \geq 90, \varphi \geq 0, |\psi| > 100$	1.92
G	$|\omega| \geq 90, \varphi \geq 0, |\psi| \leq 100$	4.73
O	$|\omega| < 90$	0.84

Table 1.2.2.1 Seven state dihedral angle classes and their frequencies.

Similar to secondary structure prediction, supervised learning has been the standard approach for dihedral angle prediction. In the literature, in addition to methods that predict continuous valued angles there are also methods that predict discrete angle classes. Note that there is no standard convention for mapping continuous valued angles to discrete angles. Therefore it may not be possible or practical to compare different prediction methods directly.

Typically artifical neural networks and support vector machines have been used for dihedral angle prediction [19-20]. The succes rates of dihedral angle class prediction methods varies according to how many angle classes are present. For example, using a 5-state torsion angle class representation (a simplified version of the definition in Table 1.2.2.1), Aydin et al. has acquired a succes rate of 84% using a hybrid classifier [21]. In another study that employs clustering, two state dihedral class prediction accuracy is obtained as 81.4%, five state dihedral class prediction as 65%, twelve state dihedral class prediction as 47% [20]. Note that the five state employed in this work is diffrent from the paper by Aydin et al. [21]. Dihedral angle prediction could be more effective for three-dimensional (3D) structure prediction than secondary structure [22]. Employing both information simultaneously could be even more useful.

1.2.3 Solvent Accesibility Prediction

Solvent accesibility designates whether each amino acid is on the surface or in internal region of the protein. Similar to dihedral angle prediction, for each amino acid, either the continuous valued solvent accesibility values or discrete solvent accessibility classes can be predicted. In the literature, the second problem has been studied more than the first one. Since accesible surface can take different values for different amino acids, as a result of a standardization procedure they are first transformed into relative solvent accesibility scores (before mapping to discrete classes) because relative solvent accesibility information is more useful for three-dimensional (3D) structure prediction than standard solvent accesibility information. To compute relative accessibility, each amino acid's accesible surface calculated by the DSSP program [23] is divided into the maximum accesibility score of that amino acid. In the next step, the relative solvent accesibility values are transformed into discrete accesibility classes. For this purpose, various number of accesibility classes have been proposed. Extensively, two, three and four classes are used. For example, in two class accesibility definition, each amino acid in the training set is assigned to exposed or buried class. For this assignment, continuous valued relative accessibility scores are compared to a threshold and are assigned to distinct classes. The threshold values are typically chosen as 0%, 5%, 10%, 25% or 50%. After labels are assigned to each amino acid, a learning model can be trained and predictions for solvent accesibility classes can be computed for a protein with unknown structure. To test the succes rate of solvent accesibility prediction, the predicted labels are compared with the true accesibility classes, which can be computed starting from the three-dimensional (3D) structure using the DSSP software [23].

In literature, neural networks have been widely used for solvent accesibility prediction [24]. Pollastri et al obtained 39.9% accuracy for ten state solvent accesibility prediction using artificial neural networks [25]. In another study, an 80% accuracy is obtained for two state solvent accesbility prediction using 25% as the threshold and 55.3% accuracy for four state prediction [16]. Figure 1.2.3.1 summarizes the two state solvent accesibility prediction problem.

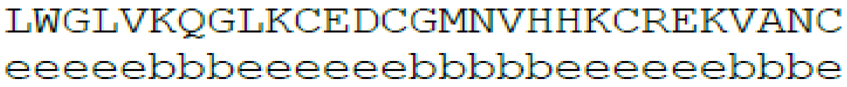

LWGLVKQGLKCEDCGMNVHHKCREKVANC
eeeeebbbeeeeeebbbbbeeeeeebbbe

Figure 1.2.3.1 Two state solvent accesibility prediction. First line is the amino acid sequence and second line is the accesibility class labels.

1.2.4 Protein Fragment Selection

Protein fragment selection aims to choose fragments that have known structure from the fragment library for each window (i.e. subsequence) of the target protein [26]. Typically overlapping windows with lengths between 1 or 20 amino acids that slide from N-terminal to C-terminal of the target protein are taken and hundreds of fragments that potentially have similar structure to the target fragment are selected (Figure 1.2.4.1).

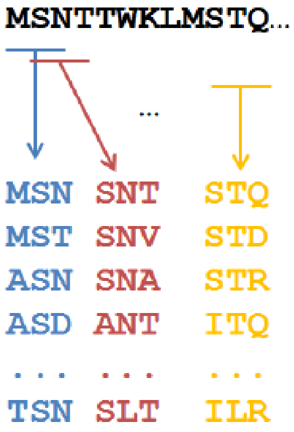


Figure 1.2.4.1 3-mer fragment selection [27].

Protein fragment selection is widely used in three-dimensional structure (3D) prediction. For three-dimensional structure (3D) prediction usually for modeling querying the Protein Data Bank (PDB) is needed. For this purpose protein fragment picker that is called fragger is used, which allows to create databases using the PDB files [28]. Methods developed for structure prediction can be grouped into two main categories: template-based modeling and template-free modeling. In template-based modeling (also known as comparative modeling), the protein structure is built by matching the target to a template protein, which can be applied when structurally related templates are available in protein structure database. When such templates cannot be found, structure prediction is performed by free modeling, which is based on the thermodynamic hypothesis that the native structure of a protein corresponds to the global minimum of its free energy for its physiological environment [29-30]. One of the popular trends among such methods is to first select a set of short fixed length fragments having 1-20 amino acids with known structures for overlapping segments of the target and then determine the tertiary structure by assembling these fragments while minimizing an energy function through statistical sampling techniques such as Monte Carlo [26]. For a succesful protein structure prediction chosing the right structural fragments plays an important role and the scoring function that is used is what determines how reliable the structure is, so optimizing the score function will yield better results for the structure prediction [31]. Protein fragment selection is also extensively used in template-based modeling, in which variable regions of the target (e.g. loops) that lack sufficient sequence similarity with the template are modeled by assembling fragments from a library of solved structures [32-33]. The most successful structure prediction methods developed to date are those that combine template-based modeling with fragment based template-free modeling.

Selection of correct (i.e., native-like) fragments is crucial for accurate prediction of 3D structure. Methods developed for fragment selection typically utilize sequence profile representations and predicted local structure such as secondary structure, torsion angles and solvent accessibility. Noise free derivation of sequence profiles and accurate prediction of local attributes are therefore important for improving the quality and accuracy of fragments and 3D models.

In the literature two main approaches have been proposed for fragment selection. The first one takes a sliding window on the target protein and aligns every amino acid sequence in the window to fragment structures in the library [30-34]. This approach is used in the most succesful 3D structure prediction methods such as I-TASSER and Rosetta. Xu and Zhang developed a gapless-threading method and showed that the optimal fragment length is 10, and at least 100 fragments are needed for optimal structure assembly [35]. In Rosetta, fragment lengths are selected as 3 and 9 [36-37]. In the second approach, fragment structures with different lengths and conformations from the protein database (PDB) are clustured [38-39]. This approach is independent from sequence information and employs structure information of the fragments. Since the sequence information is not used, the fragments can be substituted to any region of the target during a structure prediction simulation. Even though the total number of fragments selected is less compared to the first approach it could be more advantageous for fragments that have an unsteady structure and loop regions that do not have sufficient matches to fragment structures [35].

1.2.5 Protein Tertiary Structure Prediction

Many methods have been developed for predicting the 3D structure of proteins. This section will give a brief overview of these techniques. For 3D structure prediction, two of the most popular and successful methods are Rosetta and I-TASSER. Rosetta is a unified software package for protein structure prediction. Using only the sequence information, it can generate 3D models with good accuracy by assembling the fragments. Rohl et al. showed that Rosetta is among the most succesful methods for de novo protein structure prediction, which assembles fragments by a Monte Carlo strategy [40]. Simsons et al. used Rosetta to assemble 3 and 9 residue fragments by Monte Carlo simulated annealling procedure and drew conclusion from the results that ab-initio methods may soon become useful in low resolution [36]. Bradley et al. used Rosetta and methods developed since CASP5 for 3D structure prediction and obtained improved performance on large proteins [41]. Bonneau et al. used Rosetta for three-dimensional (3D) structure prediction and stated that models that were built using Rosetta were more accurate compared to the ones that were built with traditional fold recognition methods. In addition he concluded that Rosetta may be soon contribute to the interpretation of genome sequence information [42]. Yarov et al. used Rosetta for predicting helical transmembrane protein structures. Protein conformations were built by fragment assembly method of Rosetta, which predicted structure of proteins having lengths between 51 and 145 with a root-mean-square deviation < 4 Å from the original structure [43]. Gront et al. developed a new object oriented program which extends the functionality of Rosetta for fragment picking and opens new doors in protein structure modeling [37]. TASSER method is another sucessful method developed for predicting the 3D structure of proteins from the amino acid sequence. It is a hierarchical protocol and is automated. First it generates full-length atomic structural models. Then it connects to the Protein Data Bank and detects structure templates using fold recognition [44]. Wu et al. Developed the I-TASSER method, which is the iterative version of TASSER. I-TASSER was used in the folding test of three benchmarks of small proteins. The results showed that it can predict the correct folds successfully [45]. Zhang et al. used I-TASSER for protein structure prediction and obtained correct topology for 7/19 of the casses for sequences having lengths up to 155 amino acids. These results indicated that for the first time models generated by automated methods can be as good as human-experts [46]. Roy et al. developed an integrated platform for automated protein structure prediction with I-TASSER. The server outputs secondary and teritary structure predictions [47]. I-TASSER has won the CASP [48] competitions several times, which is held every two years across the globe.

In addition to I-TASSER and ROSETTA, many other methods have been proposed for 3D structure prediction. Sali et al. developed a model that uses spatial features, in which the 3D model was obtained by optimizing the molecular pdf [49]. Marti-Renom et al. showed that comperative modeling plays an important role in genome sequencing and it will play a bridging role between the two fields [50]. Ginalski et al. claims that comperative modelling is becoming a bottleneck and developing an effective all-atom-structure refinment procedures will solve this problem [51]. Ben-David et al. developed a new method called OK_RANK to evaluate the performance of template free modelling, and showed that out of 13 targets, 6 of them were predicted with high success [52].

1.3 Contributions of the Thesis

For three-dimensional (3D) structure prediction, choosing the right fragments is pretty important. Although sophisticated methods have been developed for predicting the 3D structure of proteins, many state-of-the-art methods use simple linear models to score similarity of fragments. These models have some advantages for the users. For example, some weight paramters can be assigned to zero, which disables the related score term. Furthermore, linear models can be interpreted more easily by users. This feature makes it very attractive for researchers who use Rosetta across the world. However, in a model where sequence profile information and one-dimensional structural features with different types of attributes are used, it is quite possible to have a non-linear decision boundary that separates structurally similar fragments from those that are dissimilar. Second, the weight coefficients of these linear models are typically adjusted manually or semi-automatically, which might result in models that are not optimized fully. For these reasons, using non-linear models for fragment similarity scoring, in which the weight coefficients are estimated automatically using a large dataset can potentially provide more accurate fragments.

In this thesis, machine learning methods [53] are developed for predicting whether an amino acid fragment on a target protein is structurally similar to a fragment with known structure. Two versions of the problem are studied. The first one models the structural similarity of two fragments as a classification problem, in which the output class label can be 0 or 1 depending on whether the two fragments are structurally similar or not, respectively. The second version models the similarity as a regression problem, in which the output variable can take continuous values ranging from 0 to 1 with 0 representing the most dissimilar and 1 representing identical structures. To design input features for both versions of the problem, a concept hierarchy approach is implemented, which employs PSI-BLAST's position specific scoring matrix (PSSM) (i.e. sequence profiles) [54-55], secondary structure [56], torsion angle classes [57] and solvent accessibility information [58] in various combinations and projections that summarize these features in lower dimensional spaces. After finding the best feature set representation, various classification and regression methods are trained and optimized for predicting the structural similarity of 3-mer and 9-mer fragments including logistic regression, AdaBoost, decision tree, k-nearest neighbor, naive Bayes, random forest, SVM and multi-layer perceptron. In the next step, a fragment selection method is implemented that uses the logistic regression classifier to select 200 fragment structures for each fragment window of a given target protein and its CPU performance is tested.

Chapter 2

Methods

2.1 Feature extraction and datasets for fragment selection

The fragment dataset of Rosetta (known as the vall dataset) contains 16,800 proteins and 4,126,307 amino acids, which are available in PDB (i.e. with known 3D structure). This dataset was obtained in July 2011 by Dr. Aydin from the developers of Rosetta in Baker lab. For each protein in the vall dataset, first, PSI-BLAST PSSM features and predictions of one-dimensional structural properties such as secondary structure, solvent accessibility and torsion angle classes are computed. Then train and test sets are generated by randomly sampling fragment pairs to develop machine learning models that can predict the similarity scores of two fragments.

2.1.1 PSI-BLAST PSSM features

The proteins in the vall dataset are aligned against the NR protein database using the PSI-BLAST method [59] to compute a position specific scoring matrix (PSSM), which is used as input features for the machine learning models developed in this thesis. The dimension of each PSSM is $N \times 20$, where N is the number of amino acids in the target protein and 20 represents the amino acids commonly found in nature. A PSSM represents statistical propensity scores of observing the 20 amino acids in each amino acid position of the target. The first type of PSSM is extracted by the PSI-BLAST algorithm. For this purpose, the BLAST+ program version number 2.2.31 is employed with the following parameter assignments: e-value=10^{-3}, inclusion threshold=10^{-10}, number of iterations=3. The NR database was dated as 23rd of June 2015 and contained 64,109,998 amino acid sequences. The PSI-BLAST program computes a position specific scoring matrix (PSSM) using proteins that scores above a threshold. Note that, BLAST+ was not able to find any hits for 10 proteins in the vall dataset, which are aligned aginst the NR using BLAST version 2.2.26 by setting e-value to 10, inclusion threshold to 10^{-3}, and number of iterations to 3. Although the program was still not able to find any hits, this earlier version of BLAST was able to produce PSSM features using a background model, which are used as input features for the machine learning models.

2.1.2 HHMAKE PSSM features

In addition to PSI-BLAST, the proteins in the vall dataset are also aligned against the NR20 database (a reduced version of NR) using the hhmake script of the HHsuite version 2.0.16 [60]. This script computes hidden Markov model profile (HMM-profile) for the target protein starting from the multiple alignments computed on NR20. Then the probability distributions in the match states of these HMM-profiles are converted to a PSSM. However, in our preliminary studies, the HHMAKE PSSM features did not improve the accuracy of fragment similarity estimation (results not shown). Therefore they are excluded from the feature set of the machine learning models later.

2.1.3 Predicting 1D structure using DSPRED

In addition to PSSM features, one-dimensional (1D) structural properties such as secondary structure, solvent accessibility and torsion angle classes are used as input features to estimate the similarity of two fragments. For this purpose, 1D structural properties of proteins in vall dataset are predicted using the DSPRED method, which is a two-stage hybrid classifier proposed by Aydin et al. [56]. The DSPRED method is trained on a large dataset that includes 5396 proteins derived from the PDB using the CullPDB utility of the PISCES server by setting the percentage of sequence identity threshold to 20 and removing proteins that share more than 20% sequence identity scores with the vall proteins [61]. The DSPRED method is trained separately for computing secondary structure, solvent accesssibility and torsion angle class predictions of the vall proteins. As a result of model training and prediction steps, a probability distribution of size $N \times K$ is obtained for each protein in the vall dataset, where N is the number of amino acids in the target protein and K represents the number of class labels, which is 3 for secondary structure, 2 for solvent accessibility and 7 for torsion angle classes.

Note that predictions are used as input features for the first fragment of the fragment pair only, which belongs to target protein with unknown structure. The other fragment is selected from the fragment library and has known structure. For this fragment, the true 1D structure information is coded by a probability distribution (i.e. 1-of-K scheme for each amino acid, which assigns 1 to true class and the rest is 0), which is used as input features in the machine learning models.

2.1.4 Generating train and test sets

In this thesis, a supervised learning approach is used, in which machine learning models are developed that can predict whether a pair of fragments have similar structures or not. To be able to train and validate learning models, train and test sets are generated. Because the vall fragment dataset contains billions of fragment pairs, a sampling strategy is used that randomly selects fragment pairs from the vall dataset. For this purpose, first, pairwise combinations of the 16,800 proteins are considered. Then on each protein pair, a sliding window of size 9 (i.e. 9-mer) is chosen and all possible pairwise combinations of these 9-mer fragments are considered as fragment pairs, which represent candidates for data samples in train and test sets. These candidates are divided into two pools: “candidates for similar fragment pairs” and “candidates for dissimilar fragment pairs”. The pairs that have percentage of amino acid identity score less than or equal to 50% are assigned to “candidates for dissimilar fragment pairs” pool. Otherwise if the percentage of amino acid identity score is greater than 50% and if the secondary structure labels, solvent accessibility labels and torsion angle class labels of the two fragments are identical they are assigned to “candidates for similar fragment pairs” pool. Then each pool is further divided into 26 sub-pools (a total of 52 pools) so that the tasks can be executed in different CPU cores simultaneously for faster processing. In the next step, atomic coordinates of the fragments, which are extracted from the PDB database, are compared using the BCscore program, which has been shown to capture the similarity between two structures better than the RMSD score metric [62]. The BCscore program produces a similarity score, which takes continuous values from 0 to 1 such that 0 indicates the dissimilar structures and 1 represents identical structures. This score is used as the output variable in the regression version of the fragment similarity prediction problem. For the classification version, the continuous valued BCscore output is converted to a binary class label using the following transformation

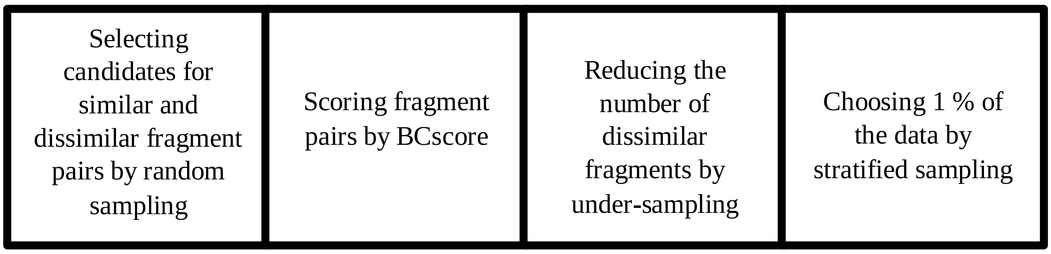
$$y = \begin{cases} 1 & BCscore > 0.63 \\ 0 & BCscore \leq 0.63 \end{cases} \quad \textbf{(2.1.4.1)}$$

where y denotes the similarity class label between the two fragments compared and 0.63 is the threshold used in the BCscore paper [62].

In order to compute the similarity between millions of fragment pairs efficiently, the C source code of the BCscore program is modified so that it receives the list of files in .pdb format for fragments that are going to be compared instead of making separate system calls to BCscore program for each fragment pair.

After labeling the fragment pairs considered, the number of pairs that are labeled as similar are much less than those that are labeled as dissimilar. Therefore to prevent class imbalance problem, a second sampling approach is employed, which performs under-sampling by selecting a subset of dissimilar fragment pairs so that the number of similar fragment pairs is nearly identical to the number of dissimilar pairs. At the end of the under-sampling procedure, 4,558,106 9-mer fragment pairs have been obtained. In the next step, 1% of these fragment pairs are further selected randomly and the size of the fragment dataset is reduced to 45,581 sample pairs. This dataset is used later in 10-fold cross-validation experiments for 9-mers.

For 3-mers the above steps are repeated. Since the number of 3-mers are much more than the number of 9-mers, an additional sampling procedure has been applied before the under-sampling step. In this procedure, a maximum of 500,000 fragment pairs are selected from each of the 52 pools obtained for 9-mers. If the number of fragment in a pool is less than 500,000 then all of these pairs are selected. Then as in 9-mers, fragment pairs selected from the 52 pools are scored simultaneously on different CPU cores using the BCscore program. At the end of these steps, 4,557,468 3-mer pairs are obtained. Finally, as in 9-mer, stratified sampling has been employed for 3-mers by selecting 1% of the fragment pairs. The fragment sampling procedures are summarized in Tables 2.1.4.1 and 2.1.4.2 for 9-mers and 3-mers, respectively.



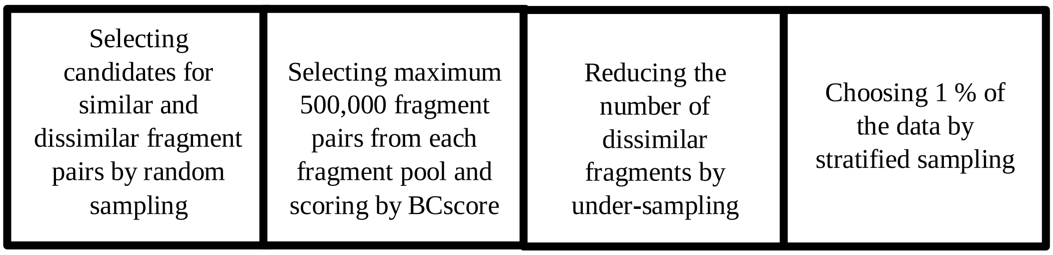
Selecting candidates for similar and dissimilar fragment pairs by random sampling

Scoring fragment pairs by BCscore

Reducing the number of dissimilar fragments by under-sampling

Choosing 1 % of the data by stratified sampling

Figure 2.1.4.1 9-mer dataset construction by sampling fragment pairs



Selecting candidates for similar and dissimilar fragment pairs by random sampling

Selecting maximum 500,000 fragment pairs from each fragment pool and scoring by BCscore

Reducing the number of dissimilar fragments by under-sampling

Choosing 1 % of the data by stratified sampling

Figure 2.1.4.2: 3-mer dataset construction by sampling fragment pairs

2.1.5 Feature vectors

Various feature parameters are considered for the feature vector of each fragment pair, which are summarized in Table 2.1.5.1.

Fragment 1			Fragment 2		
PSI-BLAST sequence profile matrices $W \times 20$ features	HHMAKE sequence profile matrices $W \times 20$ features	Local structure prediction score matrices $W \times K$ features	PSI-BLAST sequence profile matrices $W \times 20$ features	HHMAKE sequence profile matrices $W \times 20$ features	Local structure label matrices $W \times K$ features

Table 2.1.5.1 Features considered for each fragment pair

In this table, fragment 1 represents a fragment on the target protein whose structure is unknown, fragment 2 represents a fragment in the database (i.e. fragment library) whose 3D structure is known. W is the length of the fragment (i.e. number of amino acids) and is equal to 3 for 3-mers and 9 for 9-mers. K is the number of classes in structural features used, which is equal to 3 for secondary structure, 7 for torsion angle, and 2 for solvent accesibility. In fragment selection problem, since the structure of the input protein (target) is unkown the structure of fragment 1 is also considered unknown. For this reason, structural features such as secondary structure, torsion angle and solvent accesibility for the protein that fragment 1 belongs to are predicted using the DSPRED method. For example the predicted marginal probability distribution obtained for secondary structure has a size of $N \times 3$ with N denoting the number of amino acids in the protein and a size of $W \times 3$ for the fragment considered. On the other hand, since the structure of fragment 2 is known the true label information is used in hard-label format as a distribution of size $W \times K$ (i.e. orthogonal representation or 1-of-K coding scheme) in which only the true class is set to 1 and other classes are set to 0 for each amino acid in that fragment. For example, if the fragment 2 is a 3-mer and its true secondary structure labels are LEE the true label matrix takes the following form in Figure 2.1.5.1.

$$\begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{matrix}$$

Figure 2.1.5.1 An example true label matrix representing secondary structure labels of a 3-mer.

Here the rows represent the secondary structure labels (in H, E, L order), columns represent the amino acids of the fragment.

2.1.6 Feature combinations and concept hierarchy

To determine which features are useful, different combinations of the feature groups are considered. There are five feature groups: PSI-BLAST PSSMs, HHMAKE PSSMs, secondary structure distributions, torsion angle distributions and solvent accessibility distributions. Initially, the following feature groupings are implemented for these feature groups: (1) PSI-BLAST features only, (2) HHMAKE features only, (3) secondary structure features only, (4) torsion angle features only, (5) solvent accesibility features only. Then for each feature grouping, several concept hierarchies are implemented as explained below. Note that in our preliminary experiments the feature combinations that include HHMAKE PSSM features did not provide better accuracy results as compared to using PSI-BLAST PSSM features. Therefore HHMAKE PSSMs are excluded from the feature sets.

For each feature group, several concept hierarchy representations are implemented, which takes projections of the feature groups into lower dimensional spaces. Summarizing information in different levels is known as concept hierarchy in the literature of data mining [63]. This technique also reduces the overall number of dimensions (i.e. features), helping to reduce the computation time and in certain cases as in this thesis may increase the prediction accuracy of the machine learning models. In the lowest level of concept hierarchy, feature values representing fragment 1 and fragment 2 of Table 2.1.5.1 are concatenated to form a single feature vector denoted as $D0$ having a dimension of $2 \times 20 \times W$ if PSI-BLAST PSSMs are used only and $2 \times K \times W$ if structure distributions are used only. For example, if PSI-BLAST features are used only and the if the fragment size is 3, the lowest level of concept hierarchy gives $20 \times 3 = 60$ features for each fragment producing a total of 120 features. This dimension becomes $20 \times 3 \times 2 = 360$ for 9-mers. If secondary structure distributions are used only then the dimension of the feature vector $D0$ becomes $2 \times 3 \times 3 = 18$ for 3-mers and $2 \times 9 \times 3 = 54$ for 9-mers. If torsion angle class distributions are used only then the dimension of the feature vector $D0$ becomes $2 \times 7 \times 3 = 42$ for 3-mers and $2 \times 7 \times 9 = 126$ for 9-mers. If solvent accessbility class distributions are used only then the dimension of the feature vector $D0$ becomes $2 \times 3 \times 2 = 12$ for 3-mers and $2 \times 9 \times 2 = 36$ for 9-mers.

In upper levels of the concept hierarchy, distances between feature vectors are computed, which eventually is a technique used to summarize data in different dimensions. These distance scores are then concatenated to form a single feature vector for each fragment pair. Four types of distances are considered each representing a different projection of the feature matrices into a lower dimensional subspace. The first distance is computed between the PSIBLAST PSSMs of the two fragments as follows

$$D1_{psi}(i,j) = \left|M_{psi}^{(1)}(i,j) - M_{psi}^{(2)}(i,j)\right| \quad 1 \le i \le W, 1 \le j \le 20 \qquad \textbf{(2.1.6.1)}$$

where W denotes the fragment length, which is equal to 3 for 3-mers and 9 for 9-mers, $M_{psi}^{(1)}$ and $M_{psi}^{(2)}$ represent PSIBLAST PSSMs of fragments 1 and 2 respectively, both of which have dimension $W \times 20$. As it can been seen from this formula the absolute differences between the elements of the PSSM matrices are computed and a new matrix denoted as $D1_{psi}$ having a dimension of $W \times 20$ is obtained. The values in $D1_{psi}$ are then concatenated (i.e. flattened) to form a single feature vector. Dimension of this feature vector becomes 60 for 3-mers and 180 for 9-mers. These are smaller than the original dimensions, which would be 120 for 3-mers and 360 for 9-mers if the differences were not computed. The same distance formula can also be applied to structure matrices as follows

$$D1_{str}(i,j) = \left|M_{str}^{(1)}(i,j) - M_{str}^{(2)}(i,j)\right| \quad 1 \le i \le W, 1 \le j \le K \qquad \textbf{(2.1.6.2)}$$

where W denotes the fragment length, which is equal to 3 for 3-mers and 9 for 9-mers, K is the number of class labels, which is equal to 3 for secondary structure, 7 for torsion angle, 2 for solvent accessibility, str can be a structure representation such as secondary structure, torsion angle or solvent accessibility information, $M_{str}^{(1)}$ represents the predicted a posteriori probability distribution in matrix form for fragment 1 of target with a dimension of $W \times K$, $M_{str}^{(2)}$ represents the true label distribution in matrix form for fragment 2 of the library with a dimension of $W \times K$, $D1_{str}$ represents the difference matrix of dimension $W \times K$. Similar to PSIBLAST PSSM, the values in $D1_{str}$ are flattened to obtain a single feature vector, the dimension of which becomes 9 for secondary structure and 3-mer, 21 for torsion angle and 3-mer, 6 for solvent accessibility and 3-mer, 27 for secondary structure and 9-mer, 63 for torsion angle and 9-mer, 18 for solvent accessibility and 9-mer. If the differences were not computed these dimensions would be twice as high: 18 for secondary structure and 3-mer, 42 for torsion angle and 3-mer, 12 for solvent accessibility and 3-mer, 54 for secondary structure and 9-mer, 126 for torsion angle and 9-mer, 36 for solvent accessibility and 9-mer.

The second distance formula summarizes (i.e. averages) the data in position dimension of the fragments.

$$D2_{psi}(j) = \frac{1}{W}\sum_{i=1}^{W}\left|M_{psi}^{(1)}(i,j) - M_{psi}^{(2)}(i,j)\right| \quad 1 \le j \le 20 \qquad \textbf{(2.1.6.3)}$$

where $D2_{psi}$ is the distance vector, which is the summarized and normalized version of the absolute difference matrix in position dimension i and the definitions of the other terms are the same as in Equation 2.1.6.1. The dimension of $D2_{psi}$ is 20 both for 3-mer and 9-mer problems. This formula can also be applied to compute distance between structure distributions.

$$D2_{str}(j) = \frac{1}{W}\sum_{i=1}^{W}\left|M_{str}^{(1)}(i,j) - M_{str}^{(2)}(i,j)\right| \qquad 1 \le j \le K \qquad \textbf{(2.1.6.4)}$$

where $D2_{str}$ represents the difference vector of dimension K and the other terms are the same as in Equation 2.1.6.2. Dimension of this vector becomes 3 for secondary structure, 7 for torsion angle, and 2 for solvent accessibility both for 3-mer and 9-mer problems.

The third distance formula summarizes the data in the second dimension of the matrices.

$$D3_{psi}(i) = \frac{1}{20}\sum_{j=1}^{20}\left|M_{psi}^{(1)}(i,j) - M_{psi}^{(2)}(i,j)\right| \qquad 1 \le i \le W \qquad \textbf{(2.1.6.5)}$$

where $D3_{psi}$ is the distance vector, which is the summarized and normalized version of the absolute difference matrix in dimension j and the definitions of the other terms are the same as in Equation 2.1.6.1. The dimension of $D3_{psi}$ is W, which is 3 for 3-mers and 9 for 9-mers. This formula can also be applied to compute distance between structure distributions.

$$D3_{str}(i) = \frac{1}{K}\sum_{j=1}^{K}\left|M_{str}^{(1)}(i,j) - M_{str}^{(2)}(i,j)\right| \qquad 1 \le i \le W \qquad \textbf{(2.1.6.6)}$$

where $D3_{str}$ represents the difference vector of dimension W and the other terms are the same as in Equation 2.1.6.2. Dimension of this vector is 3 for 3-mers and 9 for 9-mers.

Finally the fourth distance formula takes the average in both dimensions.

$$D4_{psi} = \frac{1}{20W}\sum_{i=1}^{W}\sum_{j=1}^{20}\left|M_{psi}^{(1)}(i,j) - M_{psi}^{(2)}(i,j)\right| \qquad \textbf{(2.1.6.7)}$$

where $D4_{psi}$ is the distance term, which is the summarized and normalized version of the absolute difference matrix both in dimensions i and j and the definitions of the other terms are the same as in Equation 2.1.6.1. The dimension of $D4_{psi}$ is 1 both for 3-mers and 9-mers. This formula can also be applied to compute distance between structure distributions.

$$D4_{str} = \frac{1}{20W}\sum_{i=1}^{W}\sum_{j=1}^{K}\left|M_{str}^{(1)}(i,j) - M_{str}^{(2)}(i,j)\right| \tag{2.1.6.8}$$

Tables 2.1.6.1-2.1.6.4 summarize the number of features used for PSI-BLAST PSSMs, secondary structure, torsion angle and solvent accessibility distributions, respectively when concept hierarchy is applied using Equations (2.1.6.1)-(2.1.6.8). In these tables, the concept hierarchies are represented by D0 to D4, which correspond to the terms explained above.

Fragment Size \ Hierarchy	D0	D1	D2	D3	D4
3-mer	120	60	20	3	1
9-mer	360	180	20	9	1

Table 2.1.6.1: Number of features at different levels of concept hierarchy when PSI-BLAST PSSMs are used only to construct the feature set for 3-mers and 9-mers

Fragment Size \ Hierarchy	D0	D1	D2	D3	D4
3-mer	18	9	3	3	1
9-mer	54	27	3	9	1

Table 2.1.6.2: Number of features at different levels of concept hierarchy when secondary structure class distributions are used only to construct the feature set for 3-mers and 9-mers

Fragment Size \ Hierarchy	D0	D1	D2	D3	D4
3-mer	42	21	7	3	1
9-mer	126	63	7	9	1

Table 2.1.6.3: Number of features at different levels of concept hierarchy when torsion angle class distributions are used only to construct the feature set for 3-mers and 9-mers

Fragment Size \ Hierarchy	D0	D1	D2	D3	D4
3-mer	12	6	2	3	1
9-mer	36	18	2	9	1

Table 2.1.6.4: Number of features at different levels of concept hierarchy when solvent accessibility class distributions are used only to construct the feature set for 3-mers and 9-mers

In the concept hierarchy model described above there are a total of four feature set groups (excluding the HHMAKE PSSM features). These are PSI-BLAST PSSMs, secondary structure distributions, torsion angle class distributions, and solvent accessibility class distributions. For each feature group there can be a total of five concept hierarchy levels (D0 to D4) and two combinations originating from whether the fragment size is 3 or 9. In that case, the total number of possible feature set combinations becomes

$$\sum_{k=1}^{4} 2\binom{4}{k}5^{k} = 2390$$

Since forming 2390 different datasets for each of these combinations is computationally costly, the possible combinations are further reduced. For this purpose, first, the particular concept hierarchy level that gives the best fragment similarity prediction accuracy is found for each feature group and fragment size. Then, the following feature combinations are considered and compared using the best concept hierarchy representation for each feature group: (1) PSI-BLAST PSSM features only, (2) secondary structure distributions only, (3) torsion angle distributions only, (4) solvent accessibility distributions only, (5) PSI-BLAST and secondary structure features combined, (6) PSI-BLAST, secondary structure and torsion features combined, (7) PSI-BLAST, secondary structure, torsion and solvent accessibility features combined, (8) secondary structure, torsion and solvent accessibility features combined. For each combination, a separate dataset is prepared and a 10-fold cross-validation experiment is performed. This approach reduces the total number of feature set combinations significantly.

2.2 Prediction Methods

Two prediction problems are studied in this thesis. The first one represents the similarity between fragments as a classification problem by mapping the BCscore values to 0 or 1 (with 0 representing dissimilar and 1 representing similar fragment structures). The second one is a regression problem, which aims to predict the BCscore value directly as the output variable. Both approaches use the same feature set, which is explained in Section 2.1.

2.2.1 Classification Methods

The following classifiers are implemented to predict whether two fragments have similar 3D structures or not: logistic regression, k-nearest neighbor, decision tree, neural network, support vector machine (SVM), bagging, random forest, and AdaBoost. These methods are implemented using the WEKA software [64]. For SVM, the libSVM package is used with WEKA [65].

2.2.1.1 Logistic Regression

Logistic regression is a linear classifier, in which the decision boundary is a hyperplane. It may be attractive due to its short training times for problems that contain many numeric features and when the samples that belong to different classes can be separated by a hyperplane with high accuracy. Logistic regression can be applied both to binary and multi-class classification problems [66].

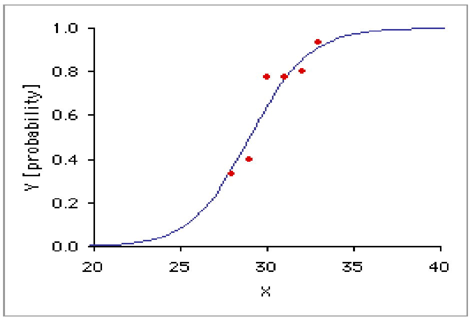


Figure 2.2.1.1.1 Logistic regression [67]

2.2.1.2 K-Nearest Neighbor

K-nearest neighbor computes the distance between the feature vector of the test sample (whose class is unknown) and the feature vectors of the train set samples. It then makes a decision by combining votes from the k samples of the train set that are closest to the test sample. For distance functions, Euclidean, Manhattan or Minkowski measures can be used [68]. In the example figure below, the green circle is classified as belonging to the red class if k is selected as 3 and to blue class if it is set to 5.

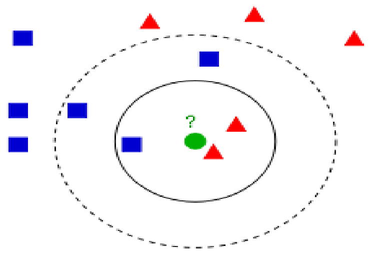


Figure 2.2.1.2.1 K-Nearest Neighbor [69]

2.2.1.3 Decision Tree

A decision tree is a supervised learning method, which starts from the root node, performing a test on an attribute at each node and makes a classification decision when it reaches to a leaf node [70]. An example desicion tree is given in Figure 2.2.1.3.1.

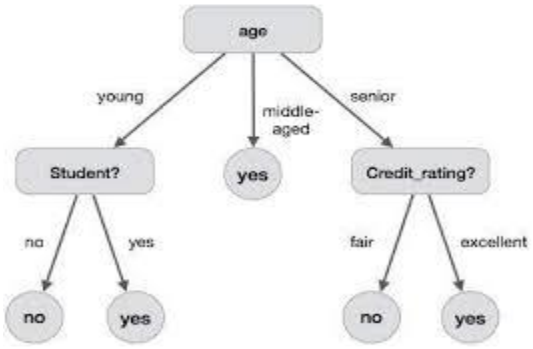


Figure 2.2.1.3.1 Decision tree [71]

2.2.1.4 Support Vector Machine

A support vector machine classifier separates the classes by a hyperplane after transforming the data to a higher dimensional space. It is among the max-margin classifiers which finds the optimum hyperplane that maximize the margin distance. Figure 2.2.1.4.1, below shows an example for a hyperplane separating two classes [72].

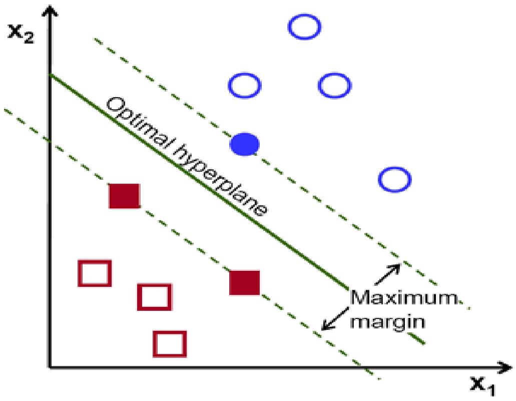


Figure 2.2.1.4.1 Suppor vector machine [73]

2.2.1.5 Artificial Neural Network

Artifical neural network is a machine learning technique which is inspired by the working principles of the human brain. It has a layered structure, where each layer contains many nodes representing neurons. Each neuron receives input signals from the previous layer and produces an output signal, which is realized by an activation function that represents whether the neuron is activated or not and to what degree. There are different types of neural networks such as feedforward and recurrent networks with feedback loops [74]. Figure 2.2.1.5.1 represents a feed-forward multi-layer perceptron network with three layers. Each edge contains a weight parameter that is multiplied with the input signal coming from the previous layer. At each hidden node (also at output node), the weighted summation of the inputs connected from the previous layer to that hidden node is computed and passed through an activation function. The activation functions at the output layer produces the output of the network.

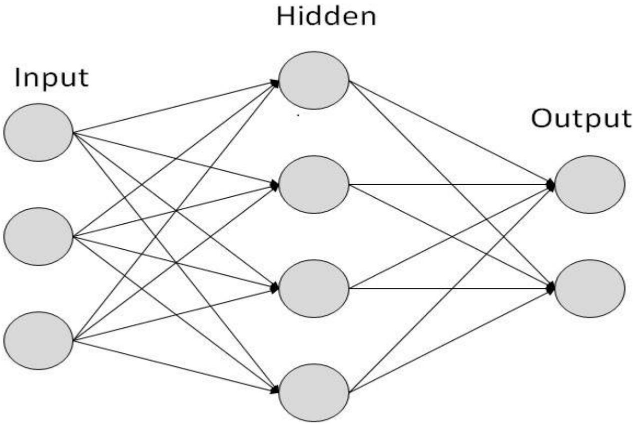


Figure 2.2.1.5.1 Artifical neural network [75]

2.2.1.6 Bagging

Bagging, also known as bootstrap aggregating, is a meta-algorithm which improves the accuracy of machine learning algorithms and is used in classification and regression. Moreover, it reduces variance and avoids overfitting. It is an ensemble learning method, in which the data is sampled with replacement from the training dataset and each bootstrapped train set is used to learn a different model. Finally the outputs from each learner is voted and a single output is obtained [76]. Figure 2.2.1.6.1 summarizes the bagging method.

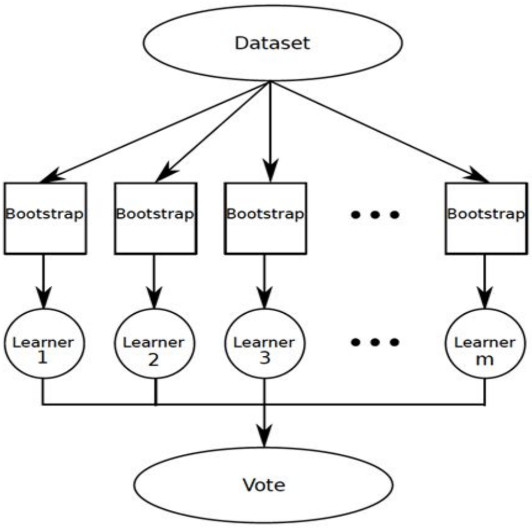


Figure 2.2.6.1.1 Bagging [77]

2.2.1.7 Random Forest

Random forest is a bagging variant, in which the base learners are decision trees each trained using a randomly selected subset of features. Random forests are robust against overfitting and outliers. They can be used both in classification and regression problems. Similar to bagging, the outputs of the base learners can be combined by a voting procedure [78].

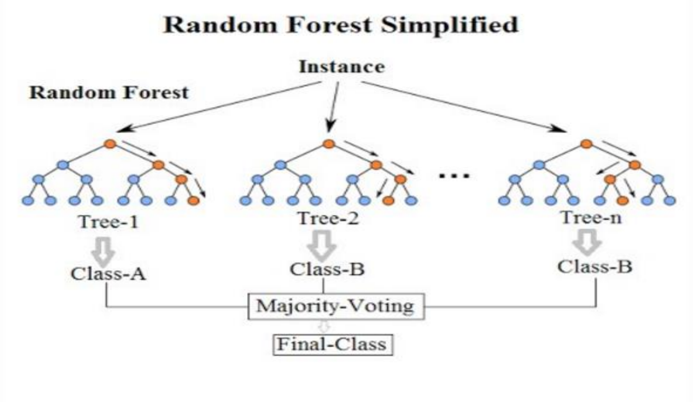


Figure 2.2.1.7.1 Random forest [79]

2.2.1.8 AdaBoost

AdaBoost is the abbreviation for “Adaptive Boosting”. It is a meta-algorithm that can improve the performance by combining several weak learners. Typically one-level desicion trees such as decision stumps are used as the base learners, which are added to the ensemble one at a time in each iteration [80]. In boosting, each train sample has a weight, which represents how likely the sample will be selected in the next iteration. Initially the weights of all the train samples are equal. After the first base learner is trained by bootstrap sampling, predictions are computed for the train samples. The weights of the samples that are misclassified are increased and the weights of those that are classified correctly are decreased. The updated weights are used to form a new bootstrap sample to train a new base learner in the next iteration. The outputs of multiple-base learners are combined by a weighted voting approach, which uses higher weights for accurate classifiers and lower weights for less accurate ones. Figure 2.2.1.8.1 summarizes the AdaBoost method.

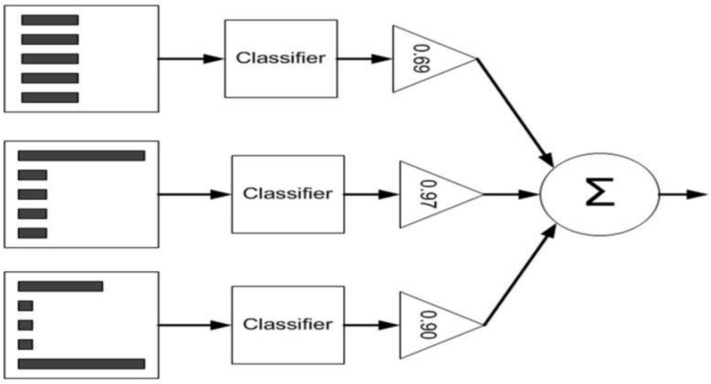


Figure 2.2.1.8.1 Adaboost [81]

2.3.1 Regression Methods

The following regression models have been implemented: linear regression, Bayesian ridge regression, MLP regression, polynomial regression, and random forest regression. The regression methods are implemented using scikit-learn library of Python [82].

2.3.1.1 Linear Regression

Linear regression fits a linear equation to the data to model the relationship between the input and output variables. The input variables are called explanatory variables and the output variable is known as response variable. Linear regression can be used to learn relations between different types of variables such as a person’s wieght and height [83].

2.3.1.2 Bayesian Ridge Regression

Bayesian ridge regression is a Bayesian approach for ridge regression. It estimates a probabilistic model of the regression problem, in which the prior distribution of the weight parameter is a spherical Gaussian [84].

2.3.1.3 MLP Regression

MLP regression uses multi layer perceptrons for regression, in which the square error is used as the loss function at the output layer [85].

2.3.1.4 Polynomial Regression

Polynomial regression is a regression model that finds a relationship between the independant and dependant variable by using a nth degree polynomial [86].

2.3.1.5 Random Forest Regression

Random forest can also be used for regression problems since decision trees can be trained both for discrete and continuous valued outputs. In scikit-learn, the predicted regression output is obtained as the mean of predictions obtained from the trees in the forest [87].

Chapter 3

Experiments and Analysis

To assess the accuracy of fragment similarity prediction, different datasets have been prepared for various feature combinations explained in Chapter 2 and for the two fragment lengths (i.e. 3-mer and 9-mer). A 10 fold cross-validation experiment is performed on each dataset. A separate validation set has been obtained from each train set by sampling 10% of the data samples randomly. These sets are used to find the optimum concept hierarchy level and to optimize the hyper-parameters of the models. The experiments and calculations has been performed using WEKA software [64]. Then models are trained on the full train set adn predictions are computed on the test set, which is repeated for the 10 folds of the cross-validation.

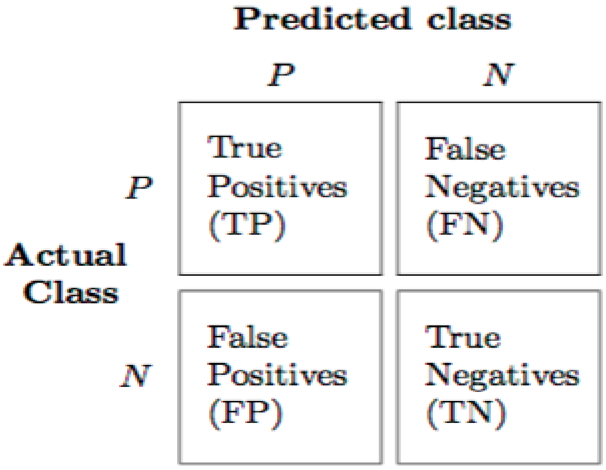
3.1 Accuracy Metrics

To evaluate the accuracy of fragment similarity classification, the following metrics have been computed: confusion matrix, overall accuracy, precision, recall, specificity, F-measure, AUC, NPV, and Matthew's correlation coefficient (MCC). For regression models, correlation, R^2 relative absolute error, root relative squared error, mean absolute error and root mean squared error metrics are computed. For the accuracy of structure predictions, the overall accuracy, recall, precision, segment overlap measure, and Matthew's correlation coefficient (MCC) are used. The definitions of these metrics are given below.

3.1.1 Accuracy Metrics for Classification

3.1.1.1 Confusion Matrix

Confusion matrix which is also known as an error matrix, is a table used to compute the accuracy of a classification algorithm [88]. Figure 3.1.1.1.1 summarizes the contents of a confusion matrix.



		Predicted class	
		P	*N*
Actual Class	*P*	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Figure 3.1.1.1.1 Confusion Matrix [89]

The definitions of the terms on this figure are given below

TP = Number of samples predicted as positive and are actually positive

TN = Number of samples predicted as negative and are actually negative

FP = Number of samples predicted as positive but are actually negative

FN = Number of samples predicted as negative but are actually positive

3.1.1.2 Overall Accuracy

To calculate the overall classification accuracy, the predictions on test sets are concatenated and compared with the correct labels. The Equation 3.1.1.2.1 formulates how this metric is computed as a percentage score.

$$AccuracyRate = 100 \times \frac{TP+TN}{TP+TN+FP+FN} \tag{3.1.1.2.1}$$

3.1.1.3 Precision

Precision is a measure for classification models, which computes how many of the positive predictions are correctly predicted as positive. It is formulated in Equation 3.1.1.3.1.

$$Precision = 100 \times \frac{TP}{TP+FP} \qquad \textbf{(3.1.1.3.1)}$$

3.1.1.4 Recall

Also known as sensitivity or true positive rate, the recall measure computes how many of the examples whose true labels are positive are correctly predicted as positive, which is formulated in Equation 3.1.1.4.1.

$$Recall = 100 \times \frac{TP}{TP+FN} \qquad \textbf{(3.1.1.4.1)}$$

3.1.1.5 Specificity

Specificity computes how many of the samples for which the true label is negative are correctly predicted as negative. Equation 3.1.1.5.1 formulates the computation of the specificity measure.

$$Specificity = 100 \times \frac{TN}{TN+FP} \qquad \textbf{(3.1.1.5.1)}$$

3.1.1.6 F-Measure

F-Measure or F-Score is another accuracy metric to measure the accuracy of a classifier. It is known as the weighted harmonic mean of recall and precision, which is calculated using the formula below.

$$F - measure = 2 \times \frac{precision\ x\ recall}{precision+recall} \qquad \textbf{(3.1.1.6.1)}$$

3.1.1.7 AUC

AUC is another accuracy metric and stands for the area under the receiver operating characteristic (ROC) curve, which is obtained as a plot of true positive rate versus false positive rate (i.e. 1-specificity) for different decision thresholds. Below in figure 3.1.1.7.1 a ROC curve is given. AUC is the area under this curve.

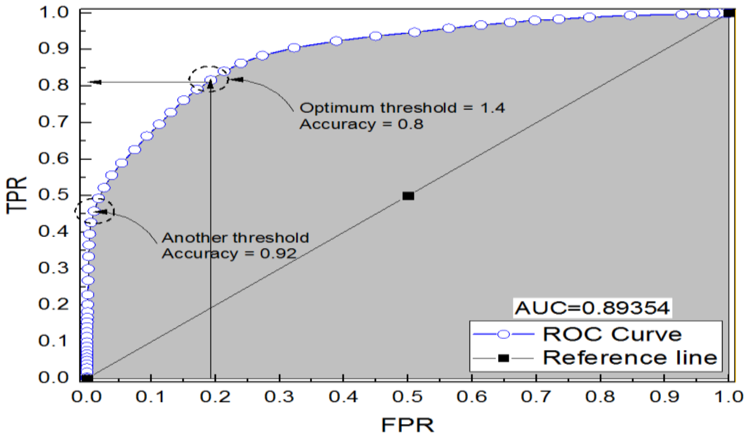


Figure 3.1.1.7.1 ROC curve

3.1.1.8 NPV

NPV which stands for the negative predictive value and is computed as how many of the negative predictions are correctly predicted as negative. Equation 3.1.1.8.1 formulates the computation of the NPV measure.

$$NPV = 100 \times \frac{TN}{TN+FN} \tag{3.1.1.8.1}$$

3.1.1.9 MCC

MCC stands for "Matthews Correlation coefficent" and is a an accuracy metric used to check the quality of a classifer. It useses the true positive, true negative, false positive and false negative values to calculate the MCC. Equation 3.1.1.9.1 shows the calculation of the MCC measure.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{3.1.1.9.1}$$

3.1.1.9 SOV

The segment overlap measure (SOV) is used to assess the accuracy of secondary structure, torsion angle class, and solvent accessibility predictions. It indicates how well the predicted structural segments for match with the true segments [90].

3.1.2 Accuracy Metrics for Regression

3.1.2.1 Correlation

Correlation is a statistical method that can show us how a pair of variables are related. Equation 3.1.2.1.1 shows the calculation of correlation measure.

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum X^2 - (\sum x^2)]\,[n\sum y^2 - (\sum y)^2]}} \tag{3.1.2.1.1}$$

3.1.2.2 R2 Score

R2 score is a statistical measure and shows us how good the data were fit to the regression line. Equation 3.1.2.2.1 formulates the calculation of R2 score. *SSres* stands for the sum of squares of residuals and *SStot* stands for the total sum of squares.

$$R^2 \equiv \frac{SS_{res}}{SS_{tot}} \tag{3.1.2.2.1}$$

3.1.2.3 Relative Absolute Error

Relative absolute error (RAE) is the difference between the aprroximation and true value as an absolute value. Formula in equation 3.1.2.3.1 has been used to calculate the relative absolute error. Actual values are the ‘a’ values and predicted values are the ‘p’ values.

$$RAE = \frac{|p_1 - a_1| + \cdots + |p_n - a_n|}{|\bar{a} - a_1| + \cdots + |\bar{a} - a_n|} \tag{3.1.2.3.1}$$

3.1.2.4 Root Relative Squared Error

Root relative squared error (RRSE) is relative to if a simple predictor has been used. The average of the actual values is a simple predictor. For root relative squared error calculation the formula in equation 3.1.2.4.1 has been used.

$$RRSE = \frac{(p_1 - a_1)^2 + \cdots + (p_n - a_n)^2}{(\bar{a} - a_1)^2 + \cdots + (\bar{a} - a_n)^2} \tag{3.1.2.4.1}$$

3.1.2.5 Mean Absolute Error

Mean absolute error (MAE) is the differnece between two continious in statistics. Mean absolute error is calculate with the given equation in 3.1.2.5.1.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \qquad \textbf{(3.1.2.5.1)}$$

3.1.2.6 Root Mean Squared Error

Root mean squared error (RMSE) is the differences between the actual values and predicted values. Equation 3.1.2.6.1 shows the formula used to calculate root mean squared error. True label is the 'y' value and 'x' value is the predicted value.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (x_i - y_i)^2}{n}} \qquad \textbf{(3.1.2.6.1)}$$

3.2 Structure prediction accuracy of DSPRED

To compute predictions of secondary structure, torsion angle and solvent accessibility classes for proteins in the vall dataset, DSPRED method is trained on a large set that includes 5396 proteins derived from the PDB (PDB-PC20) as explained in Section 2.1.3. Since DSPRED is a two-stage method it is possible to obtain predictions from both stages (i.e. DBN+Committee stage or SVM stage). For torsion angle predictions computed by the SVM stage only randomly selected 100,000 amino acid samples are employed to train the SVM model in order to reduce the computational cost. Otherwise it takes months to train a single SVM model on a dataset with 5396 proteins, which contains around a million amino acids.

Tables 3.2.1-3.2.3 summarize the secondary structure, torsion angle class and solvent accessibility prediction accuracies, respectively of DBN+Committee stage of DSPRED (i.e. DBNPRED) as well as the SVM stage (i.e. DSPRED) on the vall dataset. In this table, Q_3 corresponds to the overall accuracy, SOV is the segment overlap measure, Q_H, Q_E, and Q_L are the recall measures for helix, strand and loop, respectively,

PPV_H, PPV_E, and PPV_L are the precision measures (i.e. positive predictive values) for helix, strand and loop, respectively, MCC_H, MCC_E, and MCC_L are the Matthew's correlation coefficient values for helix, strand and loop, respectively. Even if the proteins that are similar to the vall dataset are removed from the train set, DBNPRED and DSPRED have both reasonably high prediction accuracies. DSPRED performs better than DBNPRED on vall dataset in terms of the overall accuracy by 2.18% in secondary structure prediction, 2.02% in torsion angle class prediction and 3.71% in solvent accessibility prediction.

Metric	DBNPRED-ss3	DSPRED-ss3
Q_3	83.82	86.00
SOV_3	80.10	82.56
Q_H	90.40	89.22
Q_E	79.47	81.64
Q_L	79.62	85.38
SOV_H	86.44	88.17
SOV_E	81.60	83.20
SOV_L	73.12	77.15
PPV_H	86.40	90.48
PPV_E	84.85	87.26
PPV_L	80.41	81.33
MCC_H	0.81	0.84
MCC_E	0.77	0.80
MCC_L	0.68	0.72

Table 3.2.1 Secondary structure class prediction accuracies of DBNPRED and DSPRED on vall dataset

Metric	DBNPRED-ta7	DSPRED-ta7
Q_7	73.32	75.34
SOV_7	67.77	71.41
Q_L	31.71	41.35
Q_A	92.58	90.15
Q_M	60.94	68.64
Q_B	85.17	83.40
Q_E	29.80	40.86
Q_G	51.25	57.67
Q_O	45.52	50.14
SOV_L	31.60	40.71
SOV_A	84.60	86.35
SOV_M	52.90	59.25
SOV_B	81.22	80.98
SOV_E	29.71	40.47
SOV_G	50.95	57.05
SOV_O	29.83	34.44
PPV_L	60.30	57.65
PPV_A	79.37	85.87
PPV_M	62.06	62.32
PPV_B	76.02	79.60
PPV_E	61.62	52.50
PPV_G	67.42	64.78
PPV_O	89.14	98.19
MCC_L	0.38	0.43
MCC_A	0.76	0.80
MCC_M	0.52	0.56
MCC_B	0.75	0.76
MCC_E	0.42	0.45
MCC_G	0.57	0.59
MCC_O	0.64	0.70

Table 3.2.2 Torsion angle class prediction accuracies of DBNPRED and DSPRED on vall dataset

Metric	DBNPRED-sa2	DSPRED-sa2
Q_2	77.58	81.29
SOV_2	55.99	62.34
Q_e	88.82	84.04
Q_b	66.64	78.61
SOV_e	60.91	63.58
SOV_b	51.87	61.19
PPV_e	72.15	79.26
PPV_b	85.96	83.50
MCC_e	0.57	0.63
MCC_b	0.57	0.63

Table 3.2.3 Solvent accessibility class prediction accuracies of DBNPRED and DSPRED on vall dataset

3.3 Concept hierarchy and feature combination experiments

The first set of experiments aims to find the optimum concept hierarcy and feature group combination for fragment similarity classification. For each feature set group described in Section 2.1.6, first, datasets that contain feature vectors corresponding to different concept hierarchy levels are constructed. In all concept hierarchy experiments, the structure predictions are computed using the first stage of the DSPRED method. Each dataset is split into 10 folds for 10-fold cross-validation experiment. Then 10% of samples are selected randomly from each train set as validation set. The remaining samples are used as “train set for optimization” (a total of 10 such datasets are prepared). A logistic regression classifier is trained on each “train set for optimization” and predictions are computed on the corresponding validation set. Finally the predictions obtained for the validation sets (a total of 10) are concatenated and accuracy metrics are computed by comparing the predictions with true labels. This procedure is repeated for each concept hierarchy dataset to find the best hierarchy level. Table 3.3.1. shows the accuracy metrics for the 9-mer similarity prediction on validation sets for different concept hierarchy levels and for each feature set group. In this table, psi represents PSI-BLAST PSSM, ss3 denotes 3-state secondary structure, sa2 is 2-state solvent accessibility, ta7 refers to 7-state torsion angle. The best concept hierarchy for PSI-BLAST PSSM features is D1, which contains 180 features. This is followed by D2 containing 20 features. Here the D2 level is selected because it contains significantly less number of features yielding accuracies similar to D1. As can be observed, the best concept hierarcy level computes a distance measure between the features of the fragments compared (i.e. fragment 1 and 2). The best concept hierarchy for secondary structure distribution features is D2, which includes 3 features only both for fragment 1 and fragment 2. For solvent accessibility, the best hierarcy level is obtained as D3 with 9 features and for torsion angle, D3 level gave the best validation set accuracy with 9 features. As compared to using all the features, which corresponds to D0 level, applying concept hierarchy by computing distance metrics significantly improves the accuracy of fragment similarity prediction. This improvement is 11.23% for PSI-BLAST PSSM features, 14.49% for secondary structure features, 29.68% for solvent accessibility prediction and 18.02% for torsion angle prediction all of which are significant.

Dataset	Accuracy	Fscore	AUC
psi_D0_360	77.34	79.02	85.42
psi_D1_180	88.87	89.55	95.02
psi_D2_20	88.57	89.26	94.95
psi_D3_9	85.72	86.49	92.41
psi_D4_1	85.67	86.46	92.40
ss3_D0_54	77.55	79.07	84.44
ss3_D1_27	91.98	92.63	97.63
ss3_D3_9	91.83	92.50	97.55
ss3_D2_3	92.04	92.66	97.63
ss3_D4_1	91.73	92.40	97.58
sa2_D0_36	62.15	65.25	67.79
sa2_D1_18	80.88	82.30	87.38
sa2_D3_9	91.83	92.39	95.25
sa2_D2_2	80.90	82.33	87.36
sa2_D4_1	80.90	82.33	87.36
ta7_D0_126	78.62	80.33	86.17
ta7_D1_63	91.88	92.52	97.67
ta7_D3_9	96.64	96.93	98.75
ta7_D2_7	91.93	92.56	97.63
ta7_D4_1	91.21	91.89	97.37

Table 3.3.1 Accuracies of 9-mer similarity prediction on validation sets and at different concept hierarchy levels. Feature set in each dataset is summarized by the feature type followed by the concept hierarchy level followed by the number of features.

Once the optimum concept hierarcy is found, various combinations between feature groups are formed. Table 3.3.2 summarizes the accuracy metrics of these combinations for 9-mer similarity prediction on validation sets. In this table, psi represents PSI-BLAST PSSM, ss3 denotes 3-state secondary structure, sa2 is 2-state solvent accessibility, ta7 refers to 7-state torsion angle. For PSIBLAST PSSM features, D2 level of concepy hierarchy is used, for secondary structure class distribution features D2 level, for torsion angle class distribution features D3 level and for solvent accessibility class distribution features D3 level of the hierarchy is used.

Dataset	Accuracy	FScore	AUC
psi_20	88.57	89.26	94.95
ss3_3	92.04	92.66	97.63
sa2_9	91.83	92.39	95.25
ta7_9	**96.64**	**96.93**	**98.75**
psi_20_ss3_3	95.86	96.07	98.41
psi_20_ss3_3_ta7_9	96.33	96.61	99.01
psi_20_ss3_3_ta7_9_sa2_9	**96.47**	**96.74**	**99.01**
ss3_3_ta7_9_sa2_9	93.14	93.64	98.01

Table 3.3.2 Accuracies of 9-mer similarity prediction on validation sets for different feature combinations. Feature set in each dataset is summarized by the feature type followed by the number of features.

The experiments in Table 3.3.2 are repeated for test data. For this purpose, the original (unreduced) train sets are used to train the models and predictions are computed on test sets as in regular 10-fold cross-validation. Then these predictions are concatenated and accuracy metrics are computed by comparing the predictions with true labels. Table 3.3.3 summarizes the 10-fold cross-validation accuracy results on test data for 9-mer similarity prediction and for the best concept hierarchy levels. In this table, psi represents PSI-BLAST PSSM, ss3 denotes 3-state secondary structure, sa2 is 2-state solvent accessibility, ta7 refers to 7-state torsion angle. For psi_180 D1 level, for psi_20 D2 level, for ss3_3 D2 level, for ta7_9 D3 level, for sa2_9 D3 level of concept hierarchy is used.

Dataset	Accuracy	FScore	AUC	Precision	NPV	Recall	Specificity
psi_180	88.94	89.54	95.01	89.89	87.88	89.19	88.65
psi_20	78.85	80.01	86.88	80.29	77.25	79.74	77.84
ss3_3	92.04	92.55	97.62	92.11	91.97	92.99	90.97
ta7_9	**96.71**	**96.62**	**98.71**	**96.34**	**97.15**	**97.52**	**95.80**
psi_20_ss3_3	95.77	96.01	98.73	96.19	95.31	95.84	95.70
psi_20_ss3_3_ta7_9	96.35	96.57	98.95	96.57	96.11	96.56	96.12
psi_20_ss3_3_ta7_9_sa2_9	**96.39**	**96.60**	**98.96**	**96.65**	**96.10**	**96.55**	**96.21**
ss3_3_ta7_9_sa2_9	93.38	93.75	98.11	93.98	92.70	93.52	93.21

Table 3.3.3 10-fold cross-validation accuracies of 9-mer similarity prediction on test data and at different concept hierarchy levels. Feature set in each dataset is summarized by the feature type followed by the number of features.

Based on these results, the best feature set combinations are obtained as ta7_9 with 9 torsion features at D3 level of concept hierarchy and psi_20_ss3_3_ta7_9_sa2_9 with a total of 41 features, in which D2 level is used for PSI-BLAST, D2 level for secondary structure, D2 level for torsion angle and D3 level for solvent accessibility features. Though using torsion predictions with 9 features alone has better overall accuracy, FScore, NPV and recall values, combining all feature groups has better AUC, precision and specificity. This shows that the torsion class predictions are the most useful set of features. Furthermore combining different feature sets has a positive effect on the accuracy of fragment similarity prediction.

The above experiments are repeated for 3-mer datasets. Table 3.3.4 contains the 3-mer similarity prediction accuracies on validation sets for different concept hierarchy levels. The experiment that considers D0 hierarchy for PSI-BLAST had to be repeated, which did not finish before the completion of this thesis. In this table, psi represents PSI-BLAST PSSM, ss3 denotes 3-state secondary structure, sa2 is 2-state solvent accessibility, ta7 refers to 7-state torsion angle. According Table 3.3.4, the best concept hierarchy for PSI-BLAST PSSM features is D1, which contains 60 features. This is followed by D2 containing 20 features. Here the D1 level is selected because it contains reasonable number of features. The best concept hierarchy for secondary structure distribution features is D1, which includes 9 features only both for fragment 1 and fragment 2. For solvent accessibility and torsion angle featuers, the best hierarcy level is obtained as D3 with 9 features. Again applying concept hierarchy by computing distance metrics significantly improves the accuracy of fragment similarity prediction as compared to using all the features at D0 level. This improvement is 9.42% for secondary structure features, 11.97% for solvent accessibility prediction and 16.10% for torsion angle prediction all of which are significant.

Dataset	Accuracy	FScore	AUC
psi_D1_60	**78.91**	**80.22**	**86.93**
psi_D2_20	78.44	79.72	86.62
psi_D3_3	76.81	78.13	85.07
psi_D4_1	76.71	78.03	85.06
ss3_D0_18	77.91	79.67	81.78
ss3_D1_9	**87.33**	**88.64**	**92.86**
ss3_D2_3	86.59	88.00	92.50
ss3_D3_3	86.64	88.31	92.57
ss3_D4_1	86.06	87.56	92.18
sa2_D0_12	59.57	64.23	63.45
sa2_D1_6	72.79	75.73	79.25
sa2_D3_3	**84.76**	**86.26**	**91.08**
sa2_D2_2	72.83	75.76	79.24
sa2_D4_1	72.83	75.76	79.24
ta7_D0_42	77.50	79.38	82.56
ta7_D1_21	88.04	89.06	93.60
ta7_D2_7	87.77	88.83	93.31
ta7_D3_3	**93.60**	**94.23**	**98.23**
ta7_D4_1	87.28	88.34	93.12

Table 3.3.4 Accuracies of 3-mer similarity prediction on validation sets and at different concept hierarchy levels. Feature set in each dataset is summarized by the feature type followed by the concept hierarchy level followed by the number of features.

Similar to 9-mers, after finding the optimum concept hierarcies for each feature category, various combinations between feature groups are formed for 3-mers. Table 3.3.5 summarizes the accuracy metrics of these combinations for 3-mer similarity prediction on validation sets. In this table, psi represents PSI-BLAST PSSM, ss3 denotes 3-state secondary structure, sa2 is 2-state solvent accessibility, ta7 refers to 7-state torsion angle. For PSIBLAST D1, for ss3 D1, for ta7 D3 and for sa2 D3 level of concept hierarchy is used.

Dataset	Accuracy	Fscore	AUC
psi_60	78.91	80.22	86.93
ss3_9	87.33	88.64	92.86
sa2_3	84.76	86.26	91.08
ta7_3	93.60	94.23	98.23
psi_60_ss3_9	96.16	96.47	98.95
psi_60_ss3_9_ta7_3	96.26	96.56	99.01
psi_60_ss3_9_ta7_3_sa2_3	**97.26**	**97.47**	**99.17**
ss3_9_ta7_3_sa2_3	93.31	93.81	98.02

Table 3.3.5 Accuracies of 3-mer similarity prediction on validation sets for different feature combinations. Feature set in each dataset is summarized by the feature type followed by the number of features.

The experiments in Table 3.3.5 are repeated for test data. For this purpose, the original (unreduced) train sets are used to train the models and predictions are computed on test sets as in regular 10-fold cross-validation. Then these predictions are concatenated and accuracy metrics are computed by comparing the predictions with true labels. Table 3.3.6 summarizes the 10-fold cross-validation accuracy results on test data for 3-mer similarity prediction and for the best concept hierarchy levels. In this table, psi represents PSI-BLAST PSSM, ss3 denotes 3-state secondary structure, sa2 is 2-state solvent accessibility, ta7 refers to 7-state torsion angle. For psiblast_60 D1 level, for ss3_9 D1 level, for ta7_3 D3 level, for sa2_3 D3 level of concept hierarchy is used.

Dataset_name	Accuracy	FScore	AUC	Precision	NPV	Recall	Specificity
psi_60	78.98	80.18	87.17	80.28	77.52	80.08	77.33
ss3_9	87.15	88.19	93.08	86.14	88.42	90.34	83.53
ta7_3	**94.01**	**94.49**	**98.20**	**92.47**	**95.94**	**96.60**	**91.09**
psi_60_ss3_9	91.59	92.22	96.76	90.70	92.68	93.79	89.11
psi_60_ss3_9_ta7_3	92.70	93.22	97.16	92.00	93.54	94.47	90.70
psi_60_ss3_9_ta7_3_sa2_3	**94.66**	**94.99**	**98.45**	**94.74**	**94.57**	**95.24**	**94.01**
ss3_9_ta7_3_sa2_3	89.11	89.83	95.19	89.17	89.04	90.49	87.55

Table 3.3.6 10-fold cross-validation accuracies of 3-mer similarity prediction on test data and at different concept hierarchy levels. Feature set in each dataset is summarized by the feature type followed by the number of features.

Based on these results, the optimum feature set combination is obtained as psiblast_60_ss3_9_ta7_3_sa2_3 with a total of 75 features at D1 level for PSI-BLAST, D1 level for secondary structure, D3 level for torsion angle and D3 level for solvent accessibility features. This is followed by ta7_3 with 3 torsion features at D3 level of concept hierarchy. For 3-mer similarity prediction combining different feature groups gave the best overall accuracy, F-score, AUC, precision, and specificity values while using torsion angle prediction features alone had the best NPV and recall. Similar to 9-mers, the torsion class predictions are the most useful set of features and combining different feature sets improves the accuracy of fragment similarity prediction considerably.

3.4 Fragment similarity prediction using different classifiers and regressors

3.4.1 Hyper-parameter optimization

The following hyper-parameters are optimized for the classification problem: number of nearest neighbors parameter of k-NN, C, gamma pair of SVM with RBF kernel, number of iterations in AdaBoost and Bagging, number of trees in random forest, learning rate, momentum, number of epochs and number of hidden units in multi-layer perceptron. The hyper-parameters that are optimized for the regression problem are: ridge coefficient in linear regression, number of hidden layers, number of hidden units, momentum, and learning rate in multi-layer perceptron, and number of trees in random forest. The optimizations are performed by training the models on “train set for optimization” and testing on validation sets repeatedly for 10-folds. The procedure for generating these datasets is explained in Section 3.3. The overall accuracy is optimized for classification models and the correlation metric is optimized for the regression models.

The following tables show the optimum parameters found on each validation set obtained from the train sets of the 10-fold cross-validation. In the tables below, k-NN stands for k-nearest neighbor, RF for random forest, SVM for support vector machine, MLP for multi-layer perceptron model with one hidden layer only. For classifier optimization, the I parameter in AdaBoost and Bagging stands for the number of iterations, the K parameter in k-NN represents the number of nearest neightbours, the I parameter in random forest is the number of trees, the L parameter in MLP stands for learning rate, M stands for momentum, N stands for number of epochs and H stands for the number of hidden units in an MLP with single hidden layer. For regressor optimization, K in linear regressor stands for the ridge parameter, Layer, Hunit, Momentum, and L.Rate represent number of hidden layers, number of hidden units, momentum and learning rate parameters of the multlayer perceptron, respectively, and Tree is the number of trees parameter in random forest.

For optimizing each classifer a specific range of parameters had been tested to find the optimum value for each fold. For Adaboost, the number of iterations starts from 5 and goes until 100 with increments of 5. For Bagging, the number of iterations starts from 1 and is incremented by 1 until 5. Then it is incremented by 5 until 50. The k parameter of k-NN starts from 1 and is increment by 1 until 5. Subsequently it is incremented by 5 until 60. The number of trees parameter of random forest starts from 7 and is incremented by 1 until 18. For the parameter grid of SVM, the C values are chosen as 0.03125, 0.125, 0.5, 2, 8, 32, 128, 512, 2048, 8192, 32768 and gamma values as 0.0000305176, 0.000122070, 0.000488281, 0.00195313, 0.0078125, 0.03125, 0.125, 0.5, 2, 8, 32. Then all pairwise combinations of these parameters are considered. For MLP, the number of epochs starts from 1 and is incremented by 1 until 5. This parameter is later incremented by 5 until 50. The following values are considered for the number of hidden units: 5, 10, 15, 20, 25, 50, 75, 100, 125. The momentum coefficient starts from 0.1 and goes up to 0.9 with increments of 0.1. Finally the following values are selected for the learning rate: 0.001, 0.005, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5. Similar to SVM, all combinations of these parameters are considered in the parameter grid.

For linear regression the ridge coefficient is chosen as 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000, 100000. For MLP regression epoch is taken 200, the number of hidden layers are chosen as 1, 2, 3, 4, 5. The number of hidden units starts from 3 and is incremented by 2 until 51. The momentum parameter starts from 0.1 and is incremented by 0.1 until 0.9. The learning rate is chosen as 0.5, 0.1, 0.01, 0.005, 0.001, 0.0005. For random forest regression number of trees start from 5 incremented by 5 untill 75.

	Adaboost	Bagging	K-NN	RF	SVM	MLP
Parameter	I	I	K	I	C, Gamma	L, M, N, H
Fold0	75	50	15	16	8, 0.5	0.05 ,0.1, 40, 50
Fold1	35	40	15	13	8, 0.5	0.05 ,0.1, 40, 50
Fold2	45	10	3	11	128, 0.125	0.05 ,0.1, 40, 50
Fold3	35	40	15	18	8192, 0.03125	0.05 ,0.1, 40, 50
Fold4	95	40	5	16	8192, 0.03125	0.05 ,0.1, 40, 50
Fold5	45	35	15	14	8192, 0.03125	0.05 ,0.1, 40, 50
Fold6	90	40	15	16	0.5, 0.5	0.05 ,0.1, 40, 50
Fold7	45	30	15	18	2048, 0.03125	0.05 ,0.1, 40, 50
Fold8	40	25	15	13	128, 0.125	0.05 ,0.1, 40, 50
Fold9	45	30	15	12	0.5, 0.5	0.05 ,0.1, 40, 50

Table 3.4.1.1 Optimum hyper-parameters for 9-mer fragment similarity classification on psi_20_ss3_3_ta7_9_sa2_9 dataset. Structure predictions are computed using the first stage of the DSPRED method.

	Adaboost	Bagging	K-NN	RF	SVM	MLP
Parameter	I	I	K	I	C, Gamma	L, M, N, H
Fold0	45	50	5	11	8192, 0.00195313	0.01, 0.1, 50, 15
Fold1	30	50	5	13	2048, 0.00195313	0.01, 0.1, 50, 15
Fold2	40	35	5	14	2048, 0.00195313	0.01, 0.1, 50, 15
Fold3	50	50	5	17	8192, 0.00195313	0.01, 0.1, 50, 15
Fold4	95	50	5	14	32, 0.03125	0.01, 0.1, 50, 15
Fold5	90	35	5	17	512, 0.0078125	0.01, 0.1, 50, 15
Fold6	35	30	5	13	8, 0.03125	0.01, 0.1, 50, 15
Fold7	45	35	5	13	32, 0.03125	0.01, 0.1, 50, 15
Fold8	95	40	5	17	128, 0.0078125	0.01, 0.1, 50, 15
Fold9	45	40	5	16	32, 0.03125	0.01, 0.1, 50, 15

Table 3.4.1.2 Optimum hyper-parameters for 3-mer fragment similarity classification on psi_60_ss3_9_ta7_3_sa2_3 dataset. Structure predictions are computed using the first stage of the DSPRED method.

	Adaboost	Bagging	K-NN	RF	SVM	MLP
Parameter	I	I	K	I	C, Gamma	L, M, N, H
Fold0	90	20	15	20	32, 2	0.5, 0.1, 10, 20
Fold1	15	5	20	5	2, 8	0.5, 0.1, 10, 20
Fold2	15	35	25	35	2, 8	0.5, 0.1, 10, 20
Fold3	40	10	20	10	2, 8	0.5, 0.1, 10, 20
Fold4	40	5	35	5	128, 2	0.5, 0.1, 10, 20
Fold5	35	45	25	45	2, 8	0.5, 0.1, 10, 20
Fold6	45	10	45	10	2, 8	0.5, 0.1, 10, 20
Fold7	20	15	45	15	2, 8	0.5, 0.1, 10, 20
Fold8	25	15	30	15	2, 32	0.5, 0.1, 10, 20
Fold9	20	50	45	50	2, 2	0.5, 0.1, 10, 20

Table 3.4.1.3 Optimum hyper-parameters for 9-mer fragment similarity classification on ta7_9 dataset. Structure predictions are computed using the first stage of the DSPRED method.

	Adaboost	Bagging	K-NN	RF	SVM	MLP
Parameter	I	I	K	I	C, Gamma	L, M, N, H
Fold0	5	20	25	9	512, 0.0078125	0.005, 0.1, 45, 15
Fold1	70	20	25	18	8192, 0.00195313	0.005, 0.1, 45, 15
Fold2	70	20	25	15	512, 0.0078125	0.005, 0.1, 45, 15
Fold3	65	20	25	18	32, 0.03125	0.005, 0.1, 45, 15
Fold4	65	20	25	17	8, 0.125	0.005, 0.1, 45, 15
Fold5	75	20	25	18	2, 0.125	0.005, 0.1, 45, 15
Fold6	75	20	25	15	512, 0.0078125	0.005, 0.1, 45, 15
Fold7	70	20	25	9	8, 0.03125	0.005, 0.1, 45, 15
Fold8	70	20	25	11	8, 0.03125	0.005, 0.1, 45, 15
Fold9	70	100	25	7	8192, 0.00195313	0.005, 0.1, 45, 15

Table 3.4.1.4 Optimum hyper-parameters for 3-mer fragment similarity classification on psi_60_ss3_9_ta7_3_sa2_3_ds_2 dataset. Structure predictions are computed using the second stage of the DSPRED method.

	Adaboost	Bagging	K-NN	RF	SVM	MLP
Parameter	I	I	K	I	C, Gamma	L, M, N, H
Fold0	40	25	7	7	8, 0.125	0.5, 0.1, 3, 20
Fold1	75	10	7	16	8, 0.125	0.5, 0.1, 3, 20
Fold2	55	5	7	18	32, 0.125	0.5, 0.1, 3, 20
Fold3	65	100	7	7	2048, 0.03125	0.5, 0.1, 3, 20
Fold4	60	4	7	15	32, 0.125	0.5, 0.1, 3, 20
Fold5	75	100	7	7	2048, 0.03125	0.5, 0.1, 3, 20
Fold6	75	100	7	18	8192, 0.0078125	0.5, 0.1, 3, 20
Fold7	75	15	7	16	32768, 0.0078125	0.5, 0.1, 3, 20
Fold8	65	100	7	16	8192, 0.0078125	0.5, 0.1, 3, 20
Fold9	65	15	7	16	32768, 0.0078125	0.5, 0.1, 3, 20

Table 3.4.1.5 Optimum hyper-parameters for 9-mer fragment similarity classification on psi_20_ss3_3_ta7_9_sa2_9_ds_2 dataset. Structure predictions are computed using the second stage of the DSPRED method.

	Linear Regression	MLP Regression	Random Forest Regression
Parameter	K	Layer, Hunit, Momentum, L.Rate	Tree
Fold0	0.00001	4, 47, 0.1, 0.01	75
Fold1	0.1	5, 15, 0.1, 0.005	60
Fold2	1	3, 15, 0.1, 0.005	65
Fold3	10	5, 15, 0.1, 0.005	55
Fold4	1000	5, 43, 0.1, 0.01	75
Fold5	0.00001	5, 11, 0.1, 0.005	75
Fold6	0.00001	2, 23, 0.1, 0.01	65
Fold7	0.00001	5, 17, 0.1, 0.005	55
Fold8	0.00001	3, 15, 0.1, 0.005	70
Fold9	0.00001	5, 17, 0.1, 0.005	70

Table 3.4.1.6 Optimum hyper-parameters for 3-mer fragment similarity score prediction on psi_60_ss3_9_ta7_3_sa2_3_ds_2 dataset. Structure predictions are computed using the second stage of the DSPRED method.

	Linear Regression	MLP Regression	Random Forest Regression
Parameter	K	Layer, Hunit, Momentum, L.Rate	Tree
Fold0	0.00001	4, 35, 0.1, 0.005	75
Fold1	0.00001	2, 23, 0.1, 0.01	65
Fold2	0.00001	4, 45, 0.1, 0.01	75
Fold3	0.00001	2, 37, 0.1, 0.005	45
Fold4	0.00001	3, 35, 0.1, 0.01	60
Fold5	0.00001	4, 45, 0.1, 0.005	55
Fold6	10000	5, 39, 0.1, 0.005	70
Fold7	1000	4, 31, 0.1, 0.001	70
Fold8	10000	3, 31, 0.1, 0.005	70
Fold9	0.00001	3, 31, 0.1, 0.005	70

Table 3.4.1.7 Optimum hyper-parameters for 9-mer fragment similarity score prediction on psi_20_ss3_3_ta7_9_sa2_9_ds_2 dataset. Structure predictions are computed using the second stage of the DSPRED method.

3.4.2 Performance of Classification Models

After optimizing the hyper-parameters, classification models are trained using the optimum hyper-parameter configurations on the full train sets and predictions are computed on test sets of the 10-fold cross-validation experiment. The following classification models are implemented by WEKA software: logistic regression, support vector machine (SVM), k nearest neighbour (k-NN), multi-layer perceptron (MLP), naive Bayes (NBayes), BayesNet, desicion tree (DT with J48), bagging, random forest (RF) and AdaBoost. C and gamma parameters in SVM, the number of nearest neighbours in k-NN and number of trees in random forest have been optimized as explained in Section 3.4.1. For 3-mers psiblast_60_ss3_9_ta7_3_sa2_3 is employed as the feature set combination, which contains 60 features from PSI-BLAST PSSMs at concept hierarchy D1, 9 features from secondary structure distributions at concept hierarchy D1, 3 features from torsion angle distributions at concept hierarchy D3 and 3 features from solvent accessibility distributions at concept hierarchy level D3 giving a total of 75 features. Table 3.4.2.1 includes the accuracy metrics of various classification methods trained for predicting the fragment similarity class of 3-mers. A 10-fold cross-validation experiment is performed for each classifier. Note that the structure predictions (i.e. secondary structure, torsion angle class and solvent accessibility) are computed using the first stage of the DSPRED method, which combines the output of dynamic Bayesian networks with a structural profile matrix. Except for NPV and recall measures the best accuracies are obtained by the bagging method. This is followed by MLP. Furthermore, the accuracies of logistic regression are close to the accuracy of best performing methods. This shows that most of the data samples can be separated using linear decision boundaries. Using non-linear models improves the accuracy of logistic regression by approximately 0.9% only.

	Accuracy	Fscore	AUC	Precision	NPV	Recall	Specificity	MCC0	MCC1
Adaboost	94.10	94.52	98.38	93.31	95.04	95.76	92.23	0.88	0.88
Bagging	**95.56**	**95.80**	**98.81**	**96.28**	94.76	95.32	**95.83**	**0.91**	**0.91**
BayesNet	93.26	93.70	95.80	93.02	93.53	94.39	91.98	0.86	0.86
Dtj48	94.24	94.56	92.87	94.83	93.57	94.28	94.18	0.88	0.88
k-NN	90.16	91.24	94.98	86.53	**95.42**	**96.48**	82.99	0.81	0.81
NBayes	92.57	93.13	94.31	91.49	93.90	94.84	90.00	0.85	0.85
Random forest	95.12	95.42	98.52	95.05	95.19	95.79	94.35	0.90	0.90
SVM	94.94	95.24	-	95.22	94.63	95.26	94.59	0.90	0.90
MLP	95.36	95.62	98.51	95.96	94.69	95.28	95.46	**0.91**	**0.91**
Logistic regression	94.66	94.99	98.45	94.74	94.57	95.24	94.01	0.94	0.94

Table 3.4.2.1 10-fold cross-validation accuracies of methods developed for 3-mer fragment similarity class prediction. Structure predictions are computed using the first step of the DSPRED method. psi_60_ss3_9_ta7_3_sa2_3 is used as the dataset.

For 9-mers, the following feature set combinations are employed: ta7_9, which contains a total of 9 features from torsion angle distributions at concept hierarchy D3 and psiblast_20_ss3_3_ta7_9_sa2_9, which contains 20 features from PSI-BLAST PSSMs at concept hierarchy D2, 3 features from secondary structure distributions at concept hierarchy D2, 9 features from torsion angle distributions at concept hierarchy D3 and 9 features from solvent accessibility distributions at concept hierarchy level D3 producing a total of 41 features. Table 3.4.2.2 includes the accuracy metrics of various classification methods trained for predicting the fragment similarity class of 9-mers when ta7_9 is used as the feature set. A 10-fold cross-validation experiment is performed for each classifier. The best performing methods can be listed as k-NN, bagging and decision tree.

	Accuracy	Fscore	AUC	Precision	NPV	Recall	Specificity	MCC0	MCC1
Adaboost	96.50	96.72	98.85	96.44	96.57	96.99	95.95	0.93	0.93
Bagging	97.15	97.31	**99.11**	97.70	96.54	96.91	97.42	**0.94**	**0.94**
BayesNet	95.56	95.86	97.05	95.05	96.17	96.69	94.29	0.91	0.91
DT	97.05	97.20	98.19	**97.82**	96.20	96.60	**97.56**	**0.94**	**0.94**
k-NN	**97.20**	**97.36**	98.89	97.61	96.74	97.11	97.31	**0.94**	**0.94**
Nbayes	96.61	96.83	97.06	96.42	96.83	97.23	95.91	0.93	0.93
RF	97.05	97.21	98.64	97.57	96.48	96.86	97.26	**0.94**	**0.94**
SVM	97.12	97.28	**-**	97.45	96.75	97.12	97.12	**0.94**	**0.94**
MLP	97.15	97.31	98.71	97.42	96.84	97.21	97.08	**0.94**	**0.94**
Logistic regression	96.71	96.92	98.71	96.65	**97.15**	**97.52**	95.80	0.93	0.93

Table 3.4.2.2 10-fold cross-validation accuracies of methods developed for 9-mer fragment similarity class prediction. Structure predictions are computed using the first stage of the DSPRED method. ta7_9 is used as the dataset.

Table 3.4.2.3 includes the accuracy metrics of various classification methods trained for predicting the fragment similarity class of 9-mers when psiblast_20_ss3_3_ta7_9_sa2_9 is used as the feature set, which contains 20 features from PSI-BLAST PSSMs at concept hierarchy D2, 3 features from secondary structure distributions at concept hierarchy D2, 9 features from torsion angle distributions at concept hierarchy D3 and 9 features from solvent accessibility distributions at concept hierarchy level D3 giving a total of 41 features. A 10-fold cross-validation experiment is performed for each classifier. The best performing methods can be listed as MLP, k-NN, bagging and random forest. Note that MLP is the best performing method in most of the accuracy measures.

	Accuracy	Fscore	AUC	Precision	NPV	Recall	Specificity	MCC0	MCC1
Adaboost	96.06	96.29	98.92	96.22	95.88	96.37	95.71	0.92	0.92
Bagging	96.92	97.08	**99.16**	97.84	95.92	96.34	97.59	**0.94**	**0.94**
BayesNet	96.39	96.56	97.66	97.73	94.94	95.41	97.49	0.93	0.93
DT	95.83	96.05	95.87	96.50	95.08	95.61	96.07	0.92	0.92
k-NN	96.75	96.96	98.62	96.60	**96.94**	**97.32**	96.12	0.93	0.93
Nbayes	96.27	96.45	97.37	97.55	94.89	95.37	97.28	0.93	0.93
RF	96.99	97.15	98.89	97.71	96.19	96.60	97.43	**0.94**	**0.94**
SVM	96.30	96.51	**-**	96.63	95.92	96.39	96.20	0.93	0.93
MLP	**97.19**	**97.34**	99.04	**97.96**	96.35	96.73	**97.72**	**0.94**	**0.94**
Logistic	96.39	96.60	98.96	96.65	96.10	96.55	96.21	0.93	0.93

Table 3.4.2.3 10-fold cross-validation accuracies of methods developed for 9-mer fragment similarity class prediction. Structure predictions are computed using the first stage of the DSPRED method. psi_20_ss3_3_ta7_9_sa2_9 is used as the dataset.

The structure predictions (i.e. secondary structure, torsion angle class and solvent accessibility) used to construct feature sets of the learning models in Tables 3.4.2.1- 3.4.2.3 are obtained as the outputs of the first stage of the DSPRED method. If the second stage, which employs an SVM classifier is also used then the structure predictions become more accurate approximately by 2-3%. Tables 3.4.2.4 and 3.4.2.5 include the accuracy metrics of the classifiers for 3-mers and 9-mers, respectively, when the second stage of DSPRED is used to compute structure predictions for fragment 1. For 3-mers the best performing methods are obtained as bagging, k-NN and random forest. For 9-mers, MLP, bagging, k-NN, naive Bayes and random forest performs the best. Note that as compared to Tables 3.4.2.1 and 3.4.2.3, which use the same feature set combination as Tables 3.4.2.4 and 3.4.2.5 respectively, the fragment similarity performance does not improve significantly when more accurate predictions from DSPRED are used in feature sets (i.e. when DSPRED predictions are more accurate by 2-3%).

	Accuracy	Fscore	AUC	Precision	NPV	Recall	Specificity	MCC0	MCC1
Adaboost	93.21	93.72	97.98	92.17	94.48	95.31	90.83	0.86	0.86
Bagging	**95.42**	**95.67**	**98.83**	**96.24**	94.52	95.10	**95.79**	**0.91**	**0.91**
BayesNet	93.26	93.70	95.98	93.05	93.51	94.36	92.01	0.86	0.86
DT	94.11	94.44	92.83	94.76	93.38	94.11	94.11	0.88	0.88
k-NN	88.89	90.30	96.05	84.15	96.45	**97.43**	79.22	0.79	0.79
Nbayes	90.63	91.42	92.17	88.99	92.72	93.98	86.83	0.81	0.81
RF	94.86	95.19	98.49	94.74	**95.01**	95.64	93.98	0.90	0.90
SVM	93.47	93.96	-	92.38	94.81	95.60	91.06	0.87	0.87
MLP	93.62	94.09	97.95	92.61	94.85	95.62	91.36	0.87	0.87
Logistic regression	92.96	93.50	97.79	91.75	94.45	95.32	90.29	0.86	0.86

Table 3.4.2.4 10-fold cross-validation accuracies of methods developed for 3-mer fragment similarity class prediction. Structure predictions are computed using the second stage of the DSPRED method. psi_60_ss3_9_ta7_3_sa2_3_ds_2 is used as the dataset, which includes more accurate structure predictions.

	Accuracy	Fscore	AUC	Precision	NPV	Recall	Specificity	MCC0	MCC1
Adaboost	96.44	96.65	99.02	96.69	96.16	96.61	96.26	0.93	0.93
Bagging	97.09	97.25	**99.21**	97.87	96.23	96.62	97.62	**0.94**	**0.94**
BayesNet	96.44	96.60	97.59	98.12	94.66	95.12	97.94	0.93	0.93
DT	96.07	96.29	95.84	96.60	95.48	95.98	96.17	0.92	0.92
k-NN	96.91	97.09	98.49	96.89	**96.93**	**97.30**	96.46	**0.94**	**0.94**
NBayes	96.21	96.36	97.19	**98.26**	94.06	94.53	**98.11**	0.92	0.92
RF	97.09	97.25	98.87	97.87	96.24	96.63	97.62	**0.94**	**0.94**
SVM	96.28	96.50	-	96.46	96.08	96.55	95.98	0.93	0.93
MLP	**97.18**	**97.33**	98.87	98.01	96.28	96.66	97.78	**0.94**	**0.94**
Logistic regression	96.60	96.80	98.96	96.85	96.31	96.74	96.44	0.93	0.93

Table 3.4.2.5 10-fold cross-validation accuracies of methods developed for 9-mer fragment similarity class prediction. Structure predictions are computed using the second stage of the DSPRED method. psi_20_ss3_3_ta7_9_sa2_9_ds_2 is used as the dataset, which includes more accurate structure predictions.

3.4.3 Performance of Regression Models

The fragment similarity score can also be predicted directly as a continuous variable using regression models. The following regressors are implemented: linear regression, MLP regression, polynomial regression, random forest regression and Bayesian ridge regression. Similar to Section 3.4.2, first hyper-parameters of these models are optimized on the validation sets generated from train sets. Then the models are trained using the optimum hyper-parameter configurations on the full train sets and predictions are computed on test sets of the 10-fold cross-validation experiment. C and gamma parameters in SVM, the number of trees in random forest have been optimized as explained in Section 3.4.1. For 3-mers psi_60_ss3_9_ta7_3_sa2_3_ds_2 is employed as the dataset, which contains 60 features from PSI-BLAST PSSMs at concept hierarchy D1, 9 features from secondary structure distributions at concept hierarchy D1, 3 features from torsion angle distributions at concept hierarchy D3 and 3 features from solvent accessibility distributions at concept hierarchy level D3 giving a total of 75 features. In this dataset, the structure predictions (i.e. secondary structure, torsion angle class and solvent accessibility) are computed using the second stage of the DSPRED method, which employs an SVM. Table 3.4.3.1 includes the accuracy metrics of various regression methods trained for predicting the fragment similarity score of 3-mers. A 10-fold cross-validation experiment is performed for each regressor. The best performance metrics are obtained by the random forest method. Using non-linear models improves the accuracy of linear models (e.g. linear regression) by approximately 9%.

	Correlation	R2 Score	Relative Absolute Error	Root Relative Squared Error	Mean Absolute Error	Root Mean Squared Error
Liner Regression	0.8073	0.6518	0.4592	0.5900	0.2193	0.3011
MLP Regression	0.8366	0.6996	0.3775	0.5480	0.1802	0.2796
Polynomial Regression	0.8442	0.7119	0.4083	0.5366	0.1949	0.2739
Random Forest Regression	**0.8935**	**0.7981**	**0.3004**	**0.4492**	**0.1434**	**0.2292**
Bayesian Ridge Regression	0.8077	0.6524	0.4586	0.5895	0.2190	0.3008

Table 3.4.3.1 10-fold cross-validation accuracies of methods developed for 3-mer fragment similarity score prediction. Structure predictions are computed using the second stage of the DSPRED method. psi_60_ss3_9_ta7_3_sa2_3_ds_2 is used as the dataset, which includes more accurate structure predictions..

Table 3.4.3.2 includes the performance metrics of various regression methods trained for predicting the fragment similarity score of 9-mers when psi_20_ss3_3_ta7_9_sa2_9_ds_2 is used as the dataset, which contains 20 features from PSI-BLAST PSSMs at concept hierarchy D2, 3 features from secondary structure distributions at concept hierarchy D2, 9 features from torsion angle distributions at concept hierarchy D3 and 9 features from solvent accessibility distributions at concept hierarchy level D3 giving a total of 41 features. The structure predictions (i.e. secondary structure, torsion angle class and solvent accessibility) are computed using the second stage of the DSPRED method, which employs an SVM. A 10-fold cross-validation experiment is performed for each classifier. The best performing methods are found as random forest and MLP regressor.

	Correlation	R2 Score	Relative Absolute Error	Root Relative Squared Error	Mean Absolute Error	Root Mean Squared Error
Liner Regression	0.8814	0.7769	0.3553	0.4722	0.1696	0.2410
MLP Regression	**0.9149**	**0.8370**	0.2638	**0.4036**	0.1260	**0.2060**
Polynomial Regression	0.9018	0.8132	0.3088	0.4321	0.1474	0.2205
Random Forest Regression	**0.9149**	**0.8370**	**0.2606**	**0.4036**	**0.1244**	**0.2060**
Bayesian Ridge Regression	0.8815	0.7771	0.3551	0.4720	0.1696	0.2409

Table 3.4.3.2 10-fold cross-validation accuracies of methods developed for 9-mer fragment similarity class prediction. Structure predictions are computed using the second stage of the DSPRED method. psiblast_20_ss3_3_ta7_9_sa2_9_ds_2 is used as the dataset, which includes more accurate structure predictions.

3.5 Fragment selection using fragment similarity prediction

A fragment selection method that employs the fragment similarity class prediction is implemented in C language. 66 test proteins shorter than 200 amino acids are selected from the vall dataset such that none of the 66 proteins have a percentage of sequence identity greater than 20% with the remaining proteins in vall. Then a sliding window is chosen on each target and on the remaining proteins in vall dataset. The window length is set to 3 for 3-mers and 9 for 9-mers. For each fragment window on target, feature sets are constructed for all fragment windows of the remaining proteins in vall dataset and fragment similarity class is predicted along with the prediction score (i.e. a probability score from 0 to 1) using logistic regression classifier. The fragments in vall dataset are ranked with respect to their prediction scores and the best 200 fragments are selected for each window segment of the target protein. When executed on a single CPU core, it takes approximately 20 days to select fragments for a single protein. This experiment is performed on a workstation with 128 GB RAM and Intel(R) Xeon(R) E5-2690 v4 @ 2.60GHz CPUs. The reason for this time duration is due to the multiple nested for loops and repeated commands that read protein sequences and their features from files. The fragment selection method can be made faster considerably if the for loops can be parallelized on GPU cores.

Chapter 4

Conclusions

In this thesis, classification and regression methods have been implemented and optimized for fragment similarity prediction and fragment selection. Fragment sizes are selected as 3 and 9. A concept hierarchy approach has been developed that finds the best projection of feature sets to lower dimensional subspaces. Furthermore the best feature group combination has been found. Implementing non-linear models improved the accuracy of fragment similarity prediction by 0.9% in classification and 9% in regression problem. Using 2-3% more accurate predictions for secondary structure, torsion angles and solvent accessibility did not improve the fragment similarity prediction considerably.

As a future work, first, the fragment selection method will be parallelized using CUDA language. Second, the fragment selection method developed in this thesis will be compared to the fragment selection methods in Rosetta and I-TASSER software in terms of the 3D structure prediction accuracy. For this purpose, 3D structure of the 66 test proteins will be computed by Rosetta and I-TASSER using the standard fragment selection methods available in these software and using the proposed method. Both classification and regression methods as well as linear vs non-linear models will be tested for fragment selection.

As an alternative direction, clustering-based fragment selection can also be implemented and compared to the existing method. In clustering-based approach, how the fragments in the library will match to those on the target is an open problem. The two approaches can be combined in a single model or they can be applied separately and the selected fragments can be combined. Furthermore during fragment selection the fragment windows on target are choosen independently from each other. However, windows that are close to each other or that correspond to long-range interactions between beta-strands can be correlated. The search space can be reduced further by taking this information into account.

BIBLIOGRAPHY

[1] Protein yapısı: http://tr.wikipedia.org/wiki/Protein_yapısı (15.01.2018)

[2] https://simple.wikipedia.org/wiki/Amino_acid (15.01.2018)

[3] https://wellnessadvocate.com/images/amino/Primary_Protein_Structures-250.jpg (15.01.2018)

[4] https://wellnessadvocate.com/?dgl=70872 (15.01.2018)

[5] Alfa sarmal: http://tr.wikipedia.org/wiki/Alfa_sarmal (15.01.2018)

[6] https://www.mun.ca/biology/scarr/F09-05.jpg (15.01.2018)

[7] http://tr.wikipedia.org/wiki/Beta_yaprak (15.01.2018)

[8] http://www.mun.ca/biology/scarr/Fg13_11b_revised.gif (15.01.2018)

[9] http://en.wikipedia.org/wiki/Protein_structure_prediction (15.01.2018)

[10] https://wellnessadvocate.com/images/amino/Tertiary_Protein_Structures.jpg (17.01.2018)

[11] https://upload.wikimedia.org/wikipedia/commons/thumb/9/97/Protein_backbone_PhiPsiOmega_drawing.svg/175px-Protein_backbone_PhiPsiOmega_drawing.svg.png (17.01.2018)

[12] https://upload.wikimedia.org/wikipedia/commons/thumb/d/d3/Accessible_surface.svg/280px-Accessible_surface.svg.png (17.01.2018)

[13] http://xray.bmc.uu.se/Courses/bioinformatik2003/Intro/quat_struc.jpg (17.01.2018)

[14] Cheng J., Tegge A. N., Baldi P., Machine Learning Methods for Protein Structure Prediction, IEEE Reviews in Biomedical Engineering, 1, 41-49, (2008).

[15] http://tr.wikipedia.org/wiki/Protein_ikincil_yapısı (18.01.2018)

[16] Mirabello C., Pollastri G., Porter, PaleAle 4.0: high-accuracy prediction of protein secondary structure and relative solvent accessibility, Bioinformatics Applications Note, 29(16), 2056-2058, (2013).

[17] Li D., Li T., Cong P., Xiong W., Sun J., A novel structural position-specific scoring matrix for the prediction of protein secondary structures, Bioinformatics, 28(1), 32-39, (2012).

[18] Pollastri G., Martin A. J. M., Mooney C., Vullo A., Accurate prediction of protein secondary structure and solvent accessibility by consensus combiners of sequence and structure information, BMC Bioinformatics, 8(201), (2007).

[19] Zimmermann O., Hansmann U. H., Support vector machines for prediction of dihedral angle regions, Bioinformatics, 22(24), 3009–3015, (2006).

[20] Kountouris P., Hirst J. D., Prediction of backbone dihedral angles and protein secondary structure using support vector machines, BMC Bioinformatics, 10(437), (2009).

[21] Aydin Z., Thompson J., Bilmes J., Baker D. and Noble W. S., Protein torsion angle class prediction by a hybrid architecture of Bayesian and neural networks, Proceedings of the 13th International Conference on Bioinformatics and Computational Biology (BIOCOMP'12), 477-483, (2012).

[22] Faraggi E., Yang Y., Zhou Y., Predicting continuous local structure and the effect of its substitution for secondary structure in fragment-free protein structure prediction, Structure, 17, 1515–1527, (2009).

[23] Kabsch W., Sander C., Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features, Biopolymers, 22(12), 2577–2637, (1983).

[24] Pollastri G., Baldi P., Fariselli P., Casadio R., Prediction of coordination number and relative solvent accessibility in proteins, Proteins, 47, 142–153, (2002).

[25] Faraggi E, Xue B, Zhou Y: Improving the prediction accuracy of residue solvent accessibility and real-value backbone torsion angles of proteins by guided-learning through a two-layer neural network, Proteins: Structure, Function, and Bioinformatics, 74(4), 847–856, (2009).

[26] Simons K. T., Kooperberg C., Huang E., Baker D., Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions, J. Mol. Biol., 268, 209–225, (1997).

[27] From the tubitak application form with the title “ Protein Local Structure Prediction by Rich Feature Sets and Machine Learning Methods “

[28] Berenger F, Simoncini D, Voet A, Shrestha R, Zhang Y, Fragger: a protein fragment picker for structural queries (2017)

[29] Xu D., Zhang Y., Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field, Proteins: Structure Function and Bioinformatics, 80, 1715-1735, (2012).

[30] Lee J., Kim S. Y., Joo K., Kim I., Lee J., Prediction of protein tertiary structure using PROFESY, a novel method based on fragment assembly and conformational space annealing, Proteins, 56(4), 704-714, (2004).

[31] Kevin W.DeRonne, Karypis G., Effective Optimization Algorightms for Fragment-assembly based Protein Sructure Prediction, (2007)

[32] Roy A., Kucukural A., Zhang Y., I-TASSER: a unified platform for automated protein structure and function prediction, Nature Protocols, 5(4), 725-738, (2010).

[33] Zhang J., He Z., Wang Q., Barz B., Kosztin I., Shang Y., Xu D., Prediction of protein tertiary structures using MUFOLD, Prediction of protein tertiary structures using MUFOLD, Methods Mol Biol., 815, 3-13, (2012).

[34] Rohl C. A., Strauss C. E., Misura K. M., Baker D., Protein structure prediction using Rosetta, Methods Enzymol.,383, 66–93, (2004).

[35] Xu D., Zhang Y., Toward optimal fragment generations for ab initio protein structure assembly, Proteins: Structure, Function, and Bioinformatics, 81, 229-239, (2013).

[36] Simons KT, Bonneau R, Ruczinski I, Baker D. Ab initio protein structure prediction of CASP III targets using ROSETTA. Proteins. 1999;(Suppl 3):171–176

[37] Gront D., Kulp D. W., Vernon R. M., Strauss C. E. M., Baker D., Generalized fragment picking in Rosetta: Design, protocols and applications, PLoS One, 6(8), (2011).

[38] Kolodny R., Koehl P., Guibas L., Levitt M., Small libraries of protein fragments model native protein structures accuratelyö J Mol Biol., 323, 297–307, (2002).

[39] Baeten L., Reumers J., Tur V., Stricher F., Lenaerts T., Serrano L., Rousseau F., Schymkowitz J., Reconstruction of protein backbones from the BriX collection of canonical protein fragments, PLoS Comput. Biol., 4,e1000083, (2008).

[40] Rohl CA, Strauss CE, Misura KM, Baker D. Protein structure prediction using Rosetta. Methods Enzymol. 2004;383:66–93

[41] Bradley P, Malmstrom L, Qian B, Schonbrun J, Chivian D, et al. Free modeling with Rosetta in CASP6. Proteins. 2005;61(Suppl 7):128–134.

[42] Bonneau R, Tsai J, Ruczinski I, Chivian D, Rohl C, Strauss CE, Baker D. Rosetta in CASP4: progress in ab initio protein structure prediction. Proteins. 2001;(Suppl 5):119–126.

[43] Yarov-Yarovoy V, Schonbrun J, Baker D. Multipass membrane protein structure prediction using Rosetta. Proteins. 2006;62:1010–1025.

[44] Yang, Jianyi, and Yang Zhang. “Protein Structure and Function Prediction Using I-TASSER.” Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.] 52 (2015): 5.8.1–5.815. PMC. Web. 5 Oct. 2017.

[45] Wu S, Skolnick J, Zhang Y. Ab initio modeling of small proteins by iterative TASSER simulations. BMC Biol. 2007;5:17

[46] Zhang Y. Template-based modeling and free modeling by I-TASSER in CASP7. Proteins.

[47] Roy A, Kucukural A, Zhang Y. I-TASSER: a unified platform for automated protein structure and function prediction. Nat Protoc. 2010;5(4):725–738.

[48] https://en.wikipedia.org/wiki/CASP (20.05.2018)

[49] Sali A, Blundell TL. Comparative protein modelling by satisfaction of spatial restraints. J Mol Biol. 1993;234:779–815.

[50] Marti-Renom MA, Stuart AC, Fiser A, Sanchez R, Melo F, et al. Comparative protein structure modeling of genes and genomes. Annu Rev Biophys Biomol Struct. 2000;29:291–325.

[51] Ginalski K. Comparative modeling for protein structure prediction. Curr Opin Struct Biol. 2006;16:172–177.

[52] Ben-David M, Noivirt-Brik O, Paz A, Prilusky J, Sussman JL, Levy Y. Assessment of CASP8 structure predictions for template free targets. Proteins. 2009;77(Suppl 9):50–65.

[53] Alpaydin, Ethem. Introduction to machine learning. MIT press, 2014.

[54] Stormo, Gary D., et al. "Use of the ‘Perceptron’algorithm to distinguish translational initiation sites in E. coli." Nucleic acids research 10.9 (1982): 2997-3011.

[55] Altschul S. F., Madden T. L., Schäffer A. A., Zhang J., Zhang Z., Miller W., Lipman D. J., Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res., 25(17), 3389-402, (1997).

[56] Aydin Z., Singh A., Bilmes J., Noble W. S., Learning sparse models for a dynamic Bayesian network classifier of protein secondary structure, BMC BIoinformatics, 12(154), (2011).

[57] IUPAC, Compendium of Chemical Terminology, 2nd ed. (the "Gold Book") (1997). Online corrected version: (2006–) "Torsion angle".

[58] Lee, B; Richards, FM. (1971). "The interpretation of protein structures: estimation of static accessibility". J Mol Biol. 55 (3): 379–400. doi:10.1016/0022-2836(71)90324-X. PMID 5551392.

[59] https://www.ncbi.nlm.nih.gov/books/NBK2590/ (16.05.2018)

[60] https://github.com/soedinglab/hh-suite/blob/master/CHANGES (16.05.2018)

[61] https://www.ncbi.nlm.nih.gov/pubmed/12912846 (16.05.2018)

[62] https://omictools.com/bcscore-tool (16.05.2018)

[63] Han J., Kamber M., Pei J., Data Mining: Concepts and Techniques, Third Edition, Morgan Kauffman, (2011)

[64] https://sourceforge.net/projects/weka/ (16.05.2018)

[65] https://www.csie.ntu.edu.tw/~cjlin/libsvm/ (16.05.2018)

[66] https://www.analyticsvidhya.com/blog/2015/11/beginners-guide-on-logistic-regression-in-r/ (16.05.2018)

[67] http://vassarstats.net/logreg1.html (17.05.2018)

[68] http://www.saedsayad.com/k_nearest_neighbors.htm (17.05.2018)

[69] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm (17.05.2018)

[70] https://en.wikipedia.org/wiki/Decision_tree (17.05.2018)

[71] https://www.tutorialspoint.com/data_mining/images/dm_decision_tree.jpg (17.05.2018)

[72] https://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html (17.05.2018)

[73] https://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html (17.05.2018)

[74] https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_neural_networks.htm (18.05.2018)

[75] https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_neural_networks.htm (18.05.2018)

[76] https://en.wikipedia.org/wiki/Bootstrap_aggregating (18.05.2018)

[77] http://slideplayer.com/slide/5270015/17/images/7/Bagging+(BAGGING+is+short+for+Bootstrap+AGGregatING).jpg (18.05.2018)

[78] https://en.wikipedia.org/wiki/Random_forest (18.05.2018)

[79] https://d2wh20haedxe3f.cloudfront.net/sites/default/files/random_forest_diagram_complete.png (18.05.2018)

[80] https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/ (19.05.2018)

[81] http://vivekmishra1991.github.io/assets/adaboost/schematic.jpg (19.05.2018)

[82] http://scikit-learn.org/ (19.05.2018)

[83] http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm (20.05.2018)

[84] https://stats.stackexchange.com/questions/328342/is-bayesian-ridge-regression-another-name-of-bayesian-linear-regression (20.05.2018)

[85] http://scikit-learn.org/stable/modules/neural_networks_supervised.html (21.05.2018)

[86] https://en.wikipedia.org/wiki/Polynomial_regression (20.05.2018)

[87] https://www.quora.com/How-does-random-forest-work-for-regression-1 (20.05.2018)

[88] http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/ (21.05.2018)

[89] https://rasbt.github.io/mlxtend/user_guide/evaluate/confusion_matrix_files/confusion_matrix_1.png (21.05.2018)

[90] Zemla A, Venclovas C, Fidelis K, Rost B, A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment, Proteins, vol 34, pp. 220–223, 1999.