Spring 2019

Highly Accurate Fragment Library for Protein Fold Recognition

Wessam Elhefnawy
Old Dominion University, welhe001@odu.edu

HIGHLY ACCURATE FRAGMENT LIBRARY FOR PROTEIN FOLD RECOGNITION

by

Wessam Elhefnawy
B.S. June 2004, Arab Academy for Science and Technology, Egypt
M.S. December 2011, Arab Academy for Science and Technology, Egypt

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
May 2019

Approved by:

Yaohang Li (Director)

Stephan Olariu (Member)

Mohammad Zubair (Member)

Lesley H. Greene (Member)

ABSTRACT

Highly Accurate Fragment Library for Protein Fold Recognition

Wessam Elhefnawy
Old Dominion University, 2019
Director: Dr. Yaohang Li

Proteins play a crucial role in living organisms as they perform many vital tasks in every living cell. Knowledge of protein folding has a deep impact on understanding the heterogeneity and molecular functions of proteins. Such information leads to crucial advances in drug design and disease understanding. Fold recognition is a key step in the protein structure discovery process, especially when traditional computational methods fail to yield convincing structural homologies. In this work, we present a new protein fold recognition approach using machine learning and data mining methodologies.

First, we identify a protein structural fragment library (Frag-K) composed of a set of backbone fragments ranging from 4 to 20 residues as the structural "keywords" that can effectively distinguish between major protein folds. We firstly apply randomized spectral clustering and random forest algorithms to construct representative and sensitive protein fragment libraries from a large-scale of high-quality, non-homologous protein structures available in PDB. We analyze the impacts of clustering cut-offs on the performance of the fragment libraries. Then, the Frag-K fragments are employed as structural features to classify protein structures in major protein folds defined by SCOP (Structural Classification of Proteins). Our results show that a structural dictionary with ~400 4- to 20-residue Frag-K fragments is capable of classifying major SCOP folds with high accuracy.

Then, based on Frag-k, we design a novel deep learning architecture, so-called DeepFrag-k, which identifies fold discriminative features to improve the accuracy of protein fold recognition. DeepFrag-k is composed of two stages: the first stage employs a multimodal Deep Belief Network (DBN) to predict the potential structural fragments given a sequence, represented as a fragment vector, and then the second stage uses a deep convolution neural network (CNN) to classify the fragment vectors into the corresponding folds. Our results show that DeepFrag-k yields 92.98% accuracy in predicting the top-100 most popular fragments, which can be used to generate discriminative fragment feature vectors to improve protein fold recognition.

This dissertation is dedicated to my parents, my wife, and my daughter Salma for their endless love, support, and encouragement.

ACKNOWLEDGMENTS

First, I would like to sincerely thank my advisor Dr. Yaohang Li for his guidance, motivation, time, patience, and support during my Ph.D. study. His guidance helped me through the time of my research and writing of this dissertation. I could not have imagined having a better advisor and mentor for my Ph.D. study.

Beside my advisor, I would like to thank my committee members: Dr. Stephan Olariu, Dr. Mohammad Zubair, and Dr. Lesley H. Greene, for their support and for their input to enhance the dissertation. I gratefully acknowledge the generous support of Old Dominion University Modeling and Simulation fellowship on this research.

A special appreciation to Dr. Hussein Abdel-Wahab, who, although no longer with us, continues to inspire by his example and dedication to the students he served over the course of his career. I owe Dr. Hussein Abdel-Wahab greatly for his friendship, encouragement, and support. He was always willing to listen, help, and answer questions.

Also, I want to thank my friends at Old Dominion University, I am thankful for their companionship. I want to express gratitude to Dr. Tamer Nadeem from Virginia Commonwealth University, for his constant support and encouragement throughout my Ph.D. years.

Finally, I am especially grateful to my family. Regardless of my successes or failures, they always stand by me, support me, and love me.

TABLE OF CONTENTS

Page

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER I

INTRODUCTION

The basic life processes and several vital functions occur inside cells, with the help of specialized proteins. Proteins are complex organic compounds created by chains of amino acids. A protein's chain composition, commonly referred to as the primary structure, is determined by the gene which encodes for it; the primary structure determines the protein's tertiary structure (fold), which in turn determines the protein's function. The relationship between the protein chain and structure is the result of a free energy minimization process at the molecular level, which cannot be explicitly solved just with the rules of physics and mathematics. Hence, computational techniques are usually applied to protein structure prediction.

1.1 Problem Statement

Predicting protein folds from proteins' amino acid sequences is considered a grand challenge in computational biology. The major difficulties are: (1) the space of possible protein structure conformations is extremely large; and (2) the physics of protein tertiary structural stability is not fully understood. In order to better understand the protein structure universe, protein structure domains have been classified into structural folds according to their topologies and evolutionary relationships. Protein domains in the same fold exhibit similar structural characteristics, which are uniquely different compared to the other folds. Moreover, proteins often

perform their functions using a limited number of residues, making it meaningful to find structural similarities at the level of short protein fragments. These short protein structure fragments can be treated as structural "keywords" that uniquely distinguish one fold from the others. Consequently, a set of keyword fragments forms the signature features of a fold.

This dissertation work focuses on developing a novel computational fold recognition approach. First, we attempt to identify a set of fragments that are capable of differentiating among common protein folds. Similar to the famous Google search engine in the Internet that recommends the best related websites to view when simply supplied with a few keywords (features). Second, we present a novel protein fold recognition approach. The fundamental idea is to convert a target protein sequence into structural fragments that popularly exist in protein structures, which contains highly discriminative features to distinguish the protein fold. The proposed approach allows the recognition of the protein fold of a given target protein sequence, even if the target protein does not seem to share any evolutionary relationship with another protein of known structure, and traditional fold recognition methods fail to obtain a significant model.

1.2 Contributions of This Dissertation

In this dissertation, we focus on protein fold recognition using deep learning approaches. The specific research tasks presented in this work include:

1) Generating Frag-K: we apply the randomized spectral clustering algorithm to process large-scale protein backbone fragment sets derived from the continuously growing PDB (Protein Data Bank) [1] to generate Frag-K libraries containing 4- to 20-residue protein fragments. The Frag-K libraries are used as structural features to encode protein structures. We train random forests model on Frag-K fragments to classify major SCOP (Structural Classification of Proteins) [2] folds. Our results show that, using about 400 4- to 20-residue fragments as structural key-words, one can classify major SCOP folds with high accuracy.

2) Building DeepFrag-k: we present a novel deep neural network architecture, so-called DeepFrag-k, to classify target protein sequences into known protein folds. The fundamental idea is to convert a target protein sequence into structural fragments that popularly exist in protein structures, represented as a fragment vector, which contains highly discriminative features to distinguish the protein fold. Deep-Frag-k is composed of two stages. The first stage uses a multi-modal Deep Belief Network (DBN) to fuse multiple groups of features, including sequence composition, amino acid physicochemical properties, and evolutionary information, to precisely predict potential structure fragments for a given sequence, which are represented as a fragment vector. Then, a 1-D Convolution Neural Network (CNN) is employed to classify the fragment vector into the appropriate fold. Our results show that DeepFrag-K is more accurate, sensitive, and robust than the existing methods.

1.3 Background

1.3.1 Proteins

The primary components of all living things are proteins [3], as they carry out most of the cell functions. They present the infrastructure and structural support that holds a creature together by making the chemical reactions necessary for life, and controlling gene expression. Proteins can be categorized into two categories based on their shapes in their natural environment [4], globular and non-globular. Most of the proteins are globular, while an important non-globular class of proteins is membrane proteins, whose shapes depend on the interaction with the cell membrane.

Proteins are complex molecules composed of amino acids bonded together in long chains. In nature, there are twenty amino acids [5]. Each protein chain may consist of dozens to thousands of amino acids assembled by peptide bonds. A peptide bond occurs between the nitrogen atom at the end of one amino acid and the carbon atom at the carboxyl end of another [5]. The portion of the original amino acid molecule integrated into the protein is often called a residue.

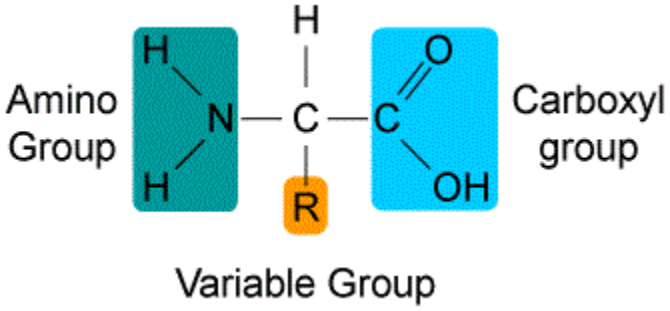


Fig. 1. The general structure of an amino acid. Reproduced from [5].

1.3.1.1 Amino Acid

Naturally, there are twenty amino acids, sharing a basic structure consisting of a central carbon atom (C), an amino group (NH3) at one end, a carboxyl group (COOH) at the other end, and a variable sidechain (R), as shown in Figure 1. The side chain determines the properties of an amino acid, where amino acids are classified based on the side chain properties [5]:

- **Polar/non-polar**: polar amino acids are the ones whose electrons are distributed asymmetrically, while non-polar ones have a relatively even distribution of charge. Some polar amino acids are positively or negatively charged in solution.
- **Hydrophobic/Hydrophilic**: hydrophobic amino acids tend to repel from water by coming together to form a compact core. Since the environment inside cells is primarily water, these hydrophobic residues tend to be on the inside of a protein, rather than on its surface.
- **Aromatic**: an aromatic amino acid forms closed rings of carbon atoms with alternating double bonds.
- **Aliphatic**: the side chain of an aliphatic amino acid side chain contains only carbon or hydrogen atoms.

Figure 2 shows a representation of the amino acids and their properties. Table 1 shows the amino acids nomenclature and comprehends alternatives.

Table 1

Nomenclature for amino acids.

Amino Acid	**Three letter code**	**One letter code**
Alanine	ALA	A
Arginine	ARG	R
Asparagine	ASN	N
Aspartic Acid	ASP	D
Cysteine	CYS	C
Glutamic Acid	GLU	E
Glutamine	GLN	Q
Glycine	GLY	G
Histidine	HIS	H
Isoleucine	ILE	I
Leucine	LEU	L
Lysine	LYS	K
Methionine	MET	M
Phenylalanine	PHE	F
Proline	PRO	P
Serine	SER	S

Threonine	THR	T
Tryptophan	TRY	W
Tyrosine	TYR	Y
Valine	VAL	V

1.3.1.2 Primary Structure

Protein's primary structure is formed by the sequence of amino acid residues. The primary structure can be represented as a sequence using the one letter code for amino acids. More general representation of the primary structure is given by profiles, which is a matrix that associates a vector to each amino acid of a protein.

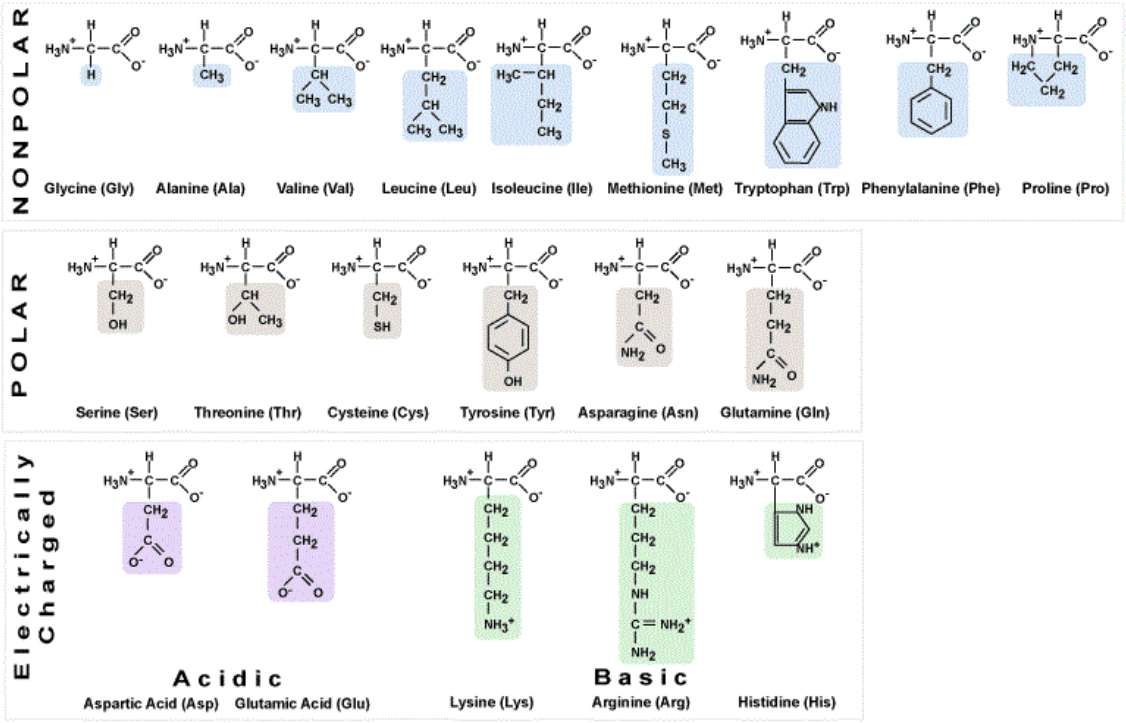


Fig. 2. Amino acid properties. Reproduced from [5].

1.3.1.3 Secondary Structure

The local conformations of amino acid residues that are seen repeatedly in proteins indicate the secondary structure [6]. Secondary structures are stabilized by hydrogen bonds. Figure 3 shows the two main kinds of secondary structure: α-helices and β-sheets (also known as β-pleated sheets). The α-helices are corkscrew-shaped conformations where the amino acids are packed tightly together. The β-sheets are made up of two or more adjacent strands of the molecule. The adjacent strands extend so that the amino acids are stretched out as far from each other as they can. Each extended chain is called a β-strand. Two or more β-strands are held together by hydrogen bonds

to form a β-sheet. There are also two main categories of β-sheet: if strands run in the same direction it is a parallel β-sheet; if they run in the opposite direction it is an anti-parallel β-sheet.

Other kinds of secondary structure are defined, as follows: The 3_{10}-helix and π-helix, are less common helix patterns. Strands formed by isolated residues are also called β-bridges. Tight turns and loose, flexible loops link the more 'regular' secondary structure elements. The conformations that are not associated with a regular secondary structure are called random loops or coils.

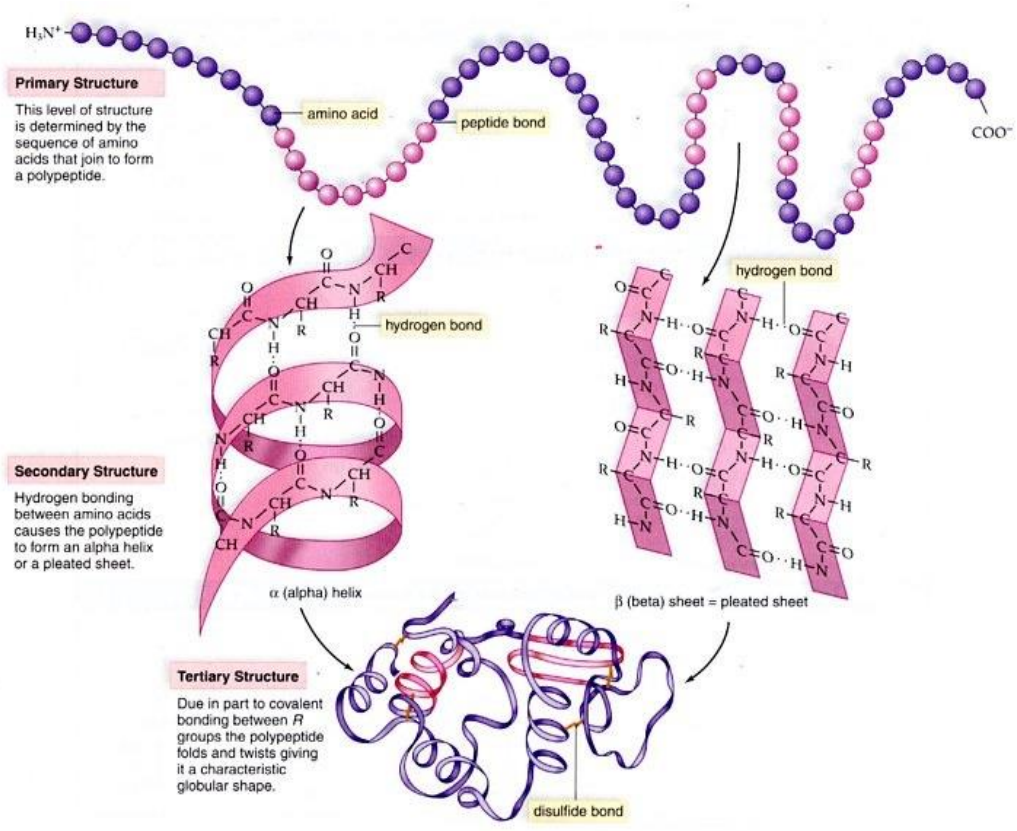


Fig. 3. Spatial arrangements of amino acid backbone occurring in α-helices and β-sheets. Reproduced from [7].

1.3.1.4 Super-Secondary Structure

It is observed in [8] that structural motifs are comprised of a few α-helices or β-strands, which are frequently repeated within structures. They are called “super-secondary structures” as they represent an intermediate structure between secondary and tertiary structures. It is suggested that these structures might be due to evolutionary convergence. A variety of recurring structures are subsequently recognized, such as the “Helix-loop-helix” and the “Greek key”, as shown in Figure 4. Some of these structural motifs can be associated with a function, while the others have no specific biological function alone, but are part of larger structural and functional assemblies.

Fig. 4. Common super secondary structure motifs.

1.3.1.5 Tertiary Structure

The three-dimensional fold of a protein is what gives them their specific chemical functionality. The link between amino acids provided by the peptide bond has two degrees of

rotational freedom, the Φ and Ψ dihedral angles. The shape when protein folds is known as the conformation of a protein backbone, which can be described as a series of Φ / Ψ angles, using the Cartesian coordinates of the central backbone atoms (the alpha carbon Ca), or using various other representational schemes. The position of the atoms in a folded protein is called its tertiary structure (Figure 5).

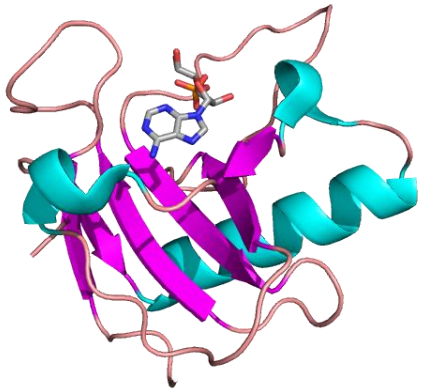


Fig. 5. Protein tertiary structure [9].

A protein's structure can be usually identified by one or more active sites that are directly associated with its functions. Some proteins bind to other proteins or groups of atoms that are required for them to function [5]. Often, several structural domains, i.e., parts of the protein that can evolve, function, and exist independently of the rest of the protein chain, can be also identified. Moreover, protein structures are not static: they can move and flex in constrained ways, which can have a significant role in their biochemical functions.

1.3.1.6 Quaternary Structure

Active conformation of multiple protein chains in one larger complex is known as the quaternary structure. A chain may bond with copies of itself or with other proteins to cooperate. Figure 6 shows an example of proteins with a quaternary structure, including DNA polymerase and ion channels.

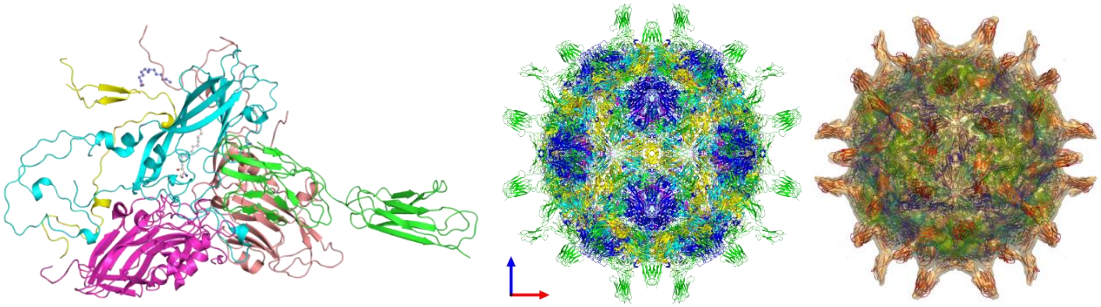


Fig. 6. Quaternary structure of viral protein, PDB id 3EPC [10].

1.3.2 From Sequence to Structure

The biological function and activity of a protein are determined from its tertiary structure [3, 4, 5], which is defined by its amino acid sequence. It is ultimately indefinite, how the properties of the amino acids in the primary structure of a protein interact to determine the protein's conformation [8]. Despite the role that amino acids properties play in protein folding, there are few absolute rules. The conformation of protein assumes the minimization of total free energy of the molecule. According to the estimation presented in [11, 12], the folding process has on average 3300 degrees of freedom. This may generate numerous alternatives, which is intractable in computer simulations. The enormous difference between the actual speed of the folding process

and the computational complexity of evaluating the corresponding model is also called Levinthal's paradox [11, 12]. Despite the development of molecular simulators that use some heuristics for reducing the search space, the uncertainty about the degree of approximation of the actual structure limits their use to only very short chains or small perturbations around a known structure. Due to the limits of molecular simulators, in most cases, a protein structure must be determined experimentally with the help of predictors.

1.3.3 Experimental Determination of Tertiary Structure

Mostly protein structures are solved experimentally using X-ray crystallography, which provides structural data of high resolution, but doesn’t give time-dependent information on the protein's conformational flexibility. Another technique to solve protein structures is NMR, which provides very high resolution data in general and is limited to relatively small proteins, but can give time-dependent information about the motion of a protein in solution. Mainly, there are more discoveries about the tertiary structural features of soluble globular proteins than about membrane proteins, because the membrane proteins are extremely difficult to study using these methods.

1.3.4 Computational Determination of Tertiary Structure

The prediction of protein tertiary structure from its amino acid sequence remains a fundamental scientific problem and it is often considered as one of the challenges in computational biology. Generally, in computational biology, five different approaches are commonly in use for

protein tertiary structure prediction. First, comparative modeling is the most accurate approach that uses experimentally clarified structures of related protein family members as templates to model the structure of a protein of interest. This approach can only be employed when a detectable template of known structure is available. Second, fold recognition and threading methods are used to model proteins that have low or statistically insignificant sequence similarity to proteins of known structure. Third, *ab initio* (*de-novo*) methods aim to predict the structure of a protein purely from its primary sequence, using principles of physics that govern protein folding and/or using information derived from known structures but without relying on any evolutionary relationship to known tertiary structures. Fourth, fragments-based methods reduce the problem to a search for the best model among a finite set of conformations. Fragments-based methods construct a complete protein structure even when it does not seem to share any relationship with a protein of known structure and traditional methods fail to obtain significant models. Finally, hybrid methods that combine information from a varied set of computational and experimental sources, including all those listed above.

1.3.4.1 Comparative Modeling

The goal of protein structure modeling, also known as homology protein structure modeling, is to build a useful tertiary structure model for a protein of unknown structure (target protein) based on one or more related proteins with known structure (templates). The most

important conditions are the detectable similarity between the target and template sequences and the possible construction of a correct alignment between them [13]. Using this approach for protein tertiary structure prediction is feasible because a slight change in the protein sequence usually only results in a slight change in its tertiary structure [13].

Comparative modeling remains the only method that can reliably predict the tertiary structure of a protein with an accuracy comparable to that of low-resolution experimental structures. Even such low-resolution models are useful to address biological questions, because the function can sometimes be predicted from only coarse structural features of a model.

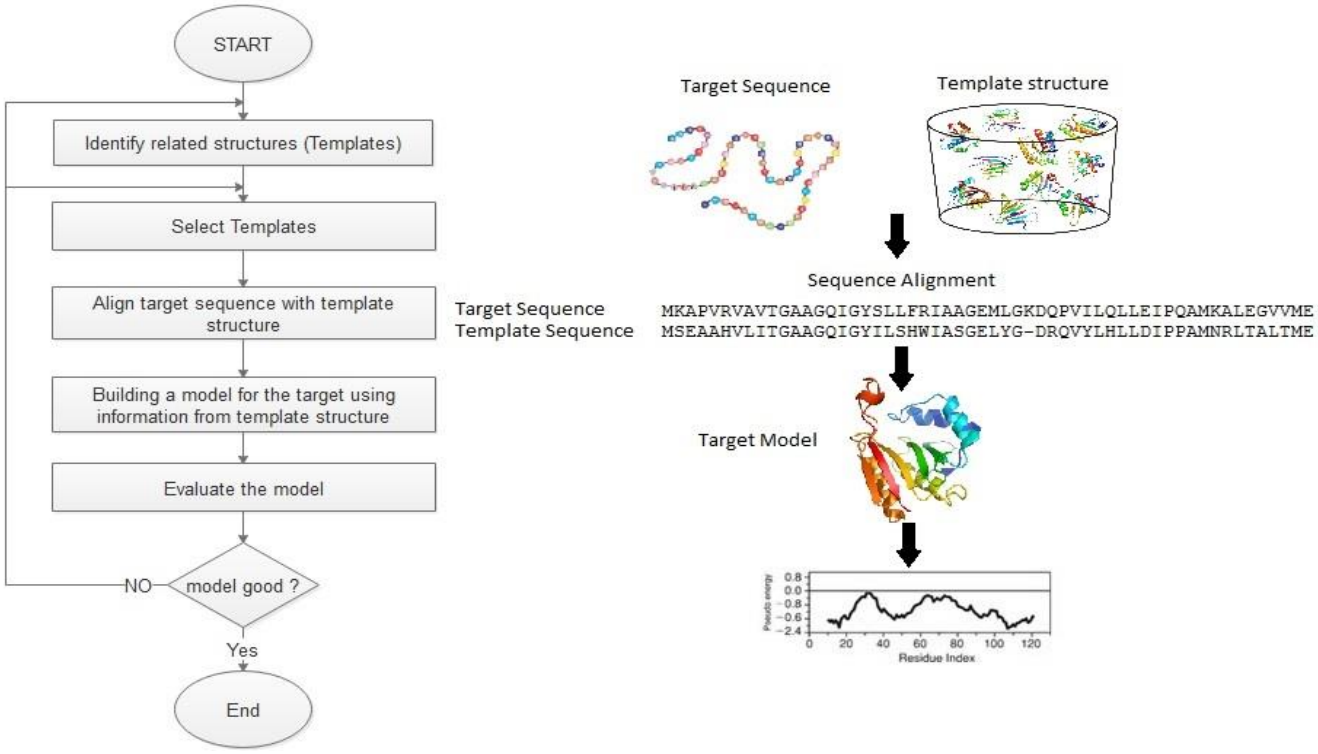


Fig. 7. Steps in comparative protein structure modeling. See text for description of each step.

As shown in Fig. *7*, comparative modeling usually consists of the following five steps: search for templates, selection of one or more templates, target-template alignment, model building, and model evaluation. If the model is not satisfactory, some or all of the steps can be repeated.

The experimental knowledge about the protein structure and its function is an important evaluation tool, where the model should be consistent with experimental observations such as site-directed mutagenesis, crosslinking data, ligand binding, etc. In cases of the best template selection and alignment are not clear, one powerful way of improving a comparative model is to change the alignment and/or the template selection and recalculate the model iteratively until no improvement in the model is detected [14]. The more exhaustive the exploration of the templates and alignments, the more likely to improve the accuracy of the final model.

1.3.4.2 Fold Recognition and Threading

Fold recognition and threading methods are used when there is no clear homology between sequences to match their tertiary structures to the target protein sequence [14]. Proteins often adopt similar folds despite even when there is no significant sequence or functional similarity [15]. Unfortunately, due to the insignificant sequence similarity, many of these fold similarities are undetected until the tertiary structure of the new protein sequence is solved. Fold recognition and

threading methods have a significant impact on protein structural biology by providing an ability to accurately identify a protein with known structures that share common tertiary structures with a target sequence. The identified tertiary structures can then be used as templates for modeling the tertiary structure of the target sequences [14]. Although these methods do not yield to equivalent models as those from experimental methods, they are faster and cheaper ways to build an approximation of a tertiary structure from a sequence [14].

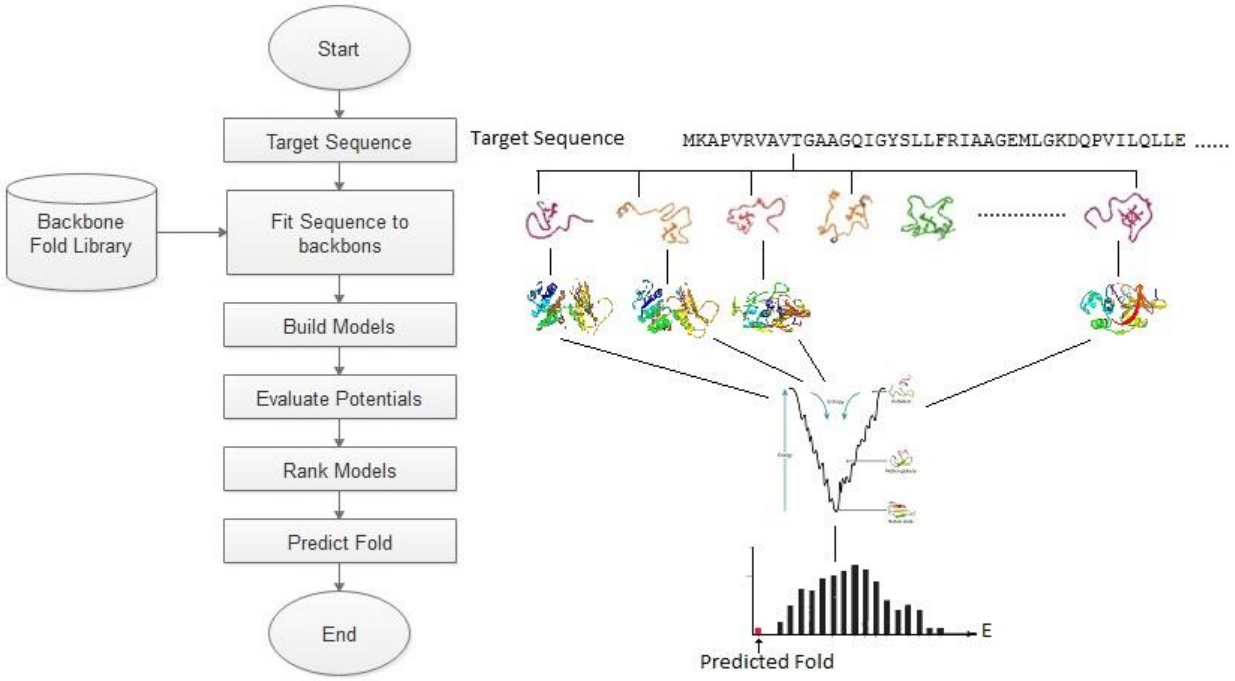


Fig. 8. A conceptual outline of fold recognition as a solution to the protein-folding problem. A given sequence (target) is fitted to the backbones of known structures (fold library), and the goodness-of-fit in each case is evaluated by one of many available model evaluation procedures (potentials).

Fold recognition and threading methods aim to assign folds to target sequences that have very low sequence identity to known structures. The original concept of early threading methods is to turn the problem of comparative modeling upside down, commonly called inverse protein folding [14]. The aim is to calculate how well each potential structure can fit a sequence, rather than how well each sequence fits a structure. In fact, fold recognition methods work by comparing each target sequence against a library of potential fold templates using energy potentials and/or similarity scoring methods. The template with the lowest energy or the highest similarity score is then assumed to best fit the fold of the target protein (Figure 8).

1.3.4.3 Ab Initio (De Novo)

In many cases, comparative modeling and fold recognition cannot provide a useful model for a target sequence, due to the lack of significant sequence similarity between the target protein sequence and a template protein sequence [14]. The chances for these methods to find a protein fold in protein structure databases increases steadily as more protein structures are solved [16]. In fact, the real problem in protein structure prediction is to know when a suitable structure is present in the PDB. In such cases, the *ab initio* methods are implemented to predict the protein secondary structure of a target sequence.

Ab initio tertiary structure prediction employs some means, which generate different protein-chain conformations and a potential function, to evaluate each conformation. Classical

force field inductive nature and knowledge-based deductive nature [17] are two different approaches that are used to obtain a potential energy function. In classical force field approaches, without previous knowledge about the protein model properties a mathematical model that describes the protein model is assumed. In these approaches, spectroscopic and thermodynamic experimental data and results from mechanical calculations in simple molecules are used to fit the adopted mathematical model. The resulting potential is directly extrapolated to more complex molecules by assuming that a common behavior exists in both cases. In knowledge-based approaches, the potential energy function of a large macro-molecular-solvent of the protein system is complex and cannot be modeled by a simple and pre-conceived mathematical model. Thus, to obtain an accurate description of the potential energy function, experimental data from large macro-molecular-solvent protein systems must be used. The potentials obtained by knowledge-based approach are called empirical potentials, statistical potentials or scoring functions.

The knowledge-based approaches do not classify types of forces, but instead, based on geometrical descriptions (i.e. distance, angles, etc.) they extract information from experimental data of known protein structures, by deriving the propensities for the interaction of two or more bodies [18]. Using principles of statistical mechanics, these approaches describe microstates of atomic interactions within protein structures as probabilities of discrete events normalized about the whole protein system. Based on the holistic nature of the knowledge-based approaches, which

accounts for atom-atom interactions as well as solvation effects, they are commonly referred to as effective energy functions. Furthermore, their strong foundations in statistical mechanics allow us to recognize a physical basis in phenomena alternative to the purely statistical one. The knowledge-based approach is not only useful for tertiary structure prediction, but also for assisting in the determination of NMR structures, where only limited data are available.

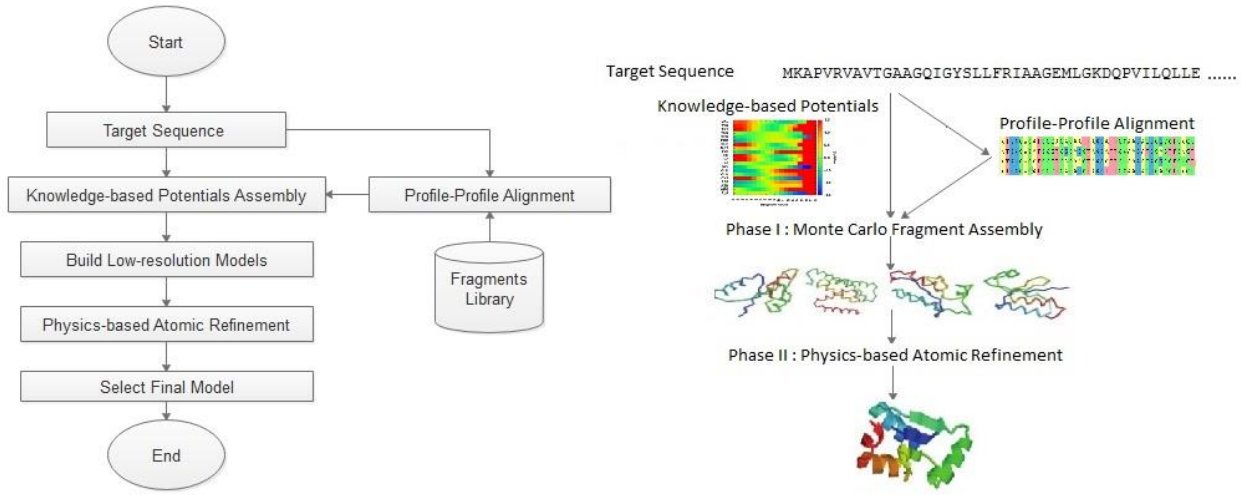


Fig. 9 ROSETTA protocol Flowchart

The knowledge-based approaches are informatics methods. Their capacity to properly describe the recurrent atomic interactions in native protein conformations depends on many parameters and on how the data are expressed and classified. In addition, the knowledge-based approaches do not only depend on how the information is extracted, expressed and classified, but also, on how the information is used. The knowledge-based approaches are widely used in protein tertiary structure prediction because of their relative simplicity, accuracy, and computational

efficiency. Among their applications, the assessment of experimentally determined and computationally predicted protein tertiary structures [13], *ab initio* protein structure prediction [17], fold recognition or threading[60], detection of native-like protein conformations [15] and prediction of protein stability [18].

Some *ab initio* methods diverge from the basic recipe described and attempt to minimize a given potential function using some simplified representation of a protein chain. Conformations of this chain can be restricted to points on a lattice [15] or restricted by choosing discrete main chain torsion angles [5, 15, 17]. Monte Carlo optimization is used, either based on some simulated annealing variants or more recently based on a genetic algorithm [18], Figure 9. Several studies are made on this aspect of protein structure prediction with some assumption differences. Although it is certainly possible to predict specific contacts in protein structures from sequences, it is difficult to use this information due to the relatively large numbers of false positives in predicted protein structures.

1.3.4.4 Fragment-Based Methods

The recently developed fold prediction methods allow the construction of a complete tertiary structure for a target protein, even when it does not seem to share any evolutionary relationship with a protein of known structure and traditional fold recognition methods fail to

obtain a significant model [19]. These new fold prediction methods are usually fragment based. They combine fragments of known structures to construct a model for a target protein.

The main idea behind the fragment-based methods is that the distribution of conformations (fragments) within a given sequence can be related to the propensity of that sequence to assume each of these conformations. Fragments with identical sequence can assume different conformations in different structures. Fragment-based protein structure prediction methods search for fragments of known structure that have a similar sequence to some fragments of the target protein and then join them together to generate a protein model. Such methods retrieve all fragments sharing some local sequence similarity with each of the fragments of the target protein and join them in many combinations. This procedure generates a large but finite set of models that can be optimized by evolutionary methods. Figure 10 shows the fragment generation protocol.

The protein folding problem is then reduced to a search for the "best" model among a given finite set of conformations, and we can use a sequence to structure score to rank the generated models. These methods raise an enormous interest because they seem to be the only current way to obtain a full tertiary structure of a protein that has no sequence or structural relationship with the set of structurally known proteins. [20].

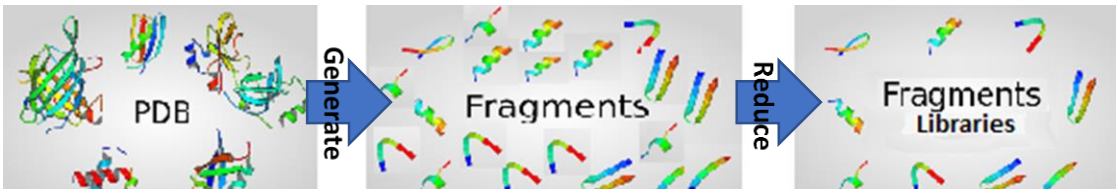


Fig. 10. Fragment Generation Protocol.

1.3.4.5 Hybrid Methods

Hybrid fully automated fold recognition servers are developed to extend the strengths of comparative modeling or fold recognition methods while limiting their weaknesses. The traditional fold recognition methods are useful for recognizing both distant homologous and analogous folds; however, they are difficult to automate and produce poor models due to inaccurate sequence to structure alignments, Fig. *11*. Alternatively, computational modeling methods are applied to extend our knowledge of protein tertiary structure, i.e. how they interact and what are their functional roles in the biological context. Frequently, the predicted protein structures are not the same as their experimental determined protein structures. Generally, the high false prediction rate comes from the need for extensive expertise to produce high-quality models and the difficulty to measure the confidence that can be associated with computationally solved structures.

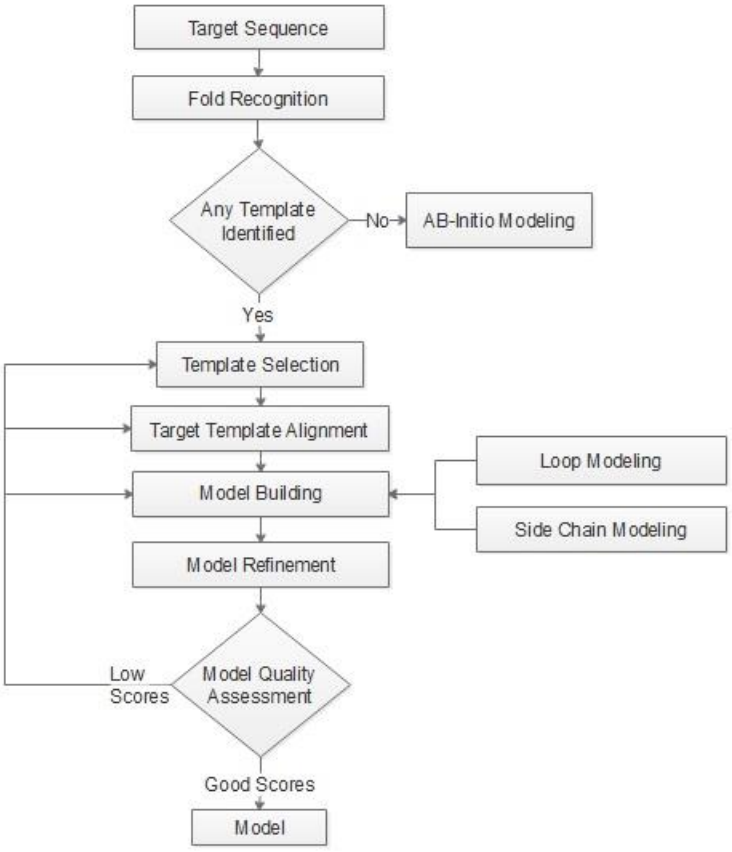


Fig. 11. Example of comparative hybrid protein tertiary modeling.

Hybrid methods aim to overcome the above weaknesses by incorporating experimental measurements, and reliable computed structural models. Hybrid approaches take advantage of data derived from a range of very different biochemical and biophysical methods, most of which are now regularly available in many laboratories. These methods are of increasing interest in view of the increasing easiness in accessing analytical instruments, such as high-resolution mass spectrometers and high-frequency electron paramagnetic resonance EPR spectrometers. Similarly, small angle neutron scattering and small angle X-ray scattering data become routinely accessible

through advanced neutron and synchrotron light sources. Recently large protein systems are made amenable to analyze due to the developments in NMR. The combination with site-specific isotope labeling opens unprecedented possibilities to obtain sparse structural data on selected regions within an entire system. Moreover, hybrid approaches show great promise in complementing high-resolution structural biology. To fully characterize the function in dynamically interacting assemblies where both the components and their structures may vary throughout a complex multistep process, structures need to be determined at each step. By using structural models, it is possible to design and analyze new hypothesis-driven experiments and thus significantly speed up high-resolution structure determination.

1.4 Dissertation Organization

The rest of this dissertation is organized as follows: Chapter 2, focuses on protein fold recognition resource and methods; Chapter 3, presents our generated large-scale protein fragment libraries Frag-K; Chapter 4, presents our two-stages deep neural network (DeepFrag-k) to classify a target protein sequence into known protein folds; Chapter 5, concludes the dissertation and discusses our future (post-dissertation) research directions.

CHAPTER II

MACHINE LEARNING FOR PROTEIN FOLD RECOGNITION

Experimental and theoretical studies lead to the emergence of a unified general mechanism for protein folding that serves as a framework for the design and interpretation of research in this area [14]. In consequence, the starting point is mainly based on some knowledge of protein folding to understand the heterogeneity and molecular function of proteins. Accordingly, computational recognition of protein folds becomes a hotspot in bioinformatics and computational biology research. Many computational efforts lead to a variety of computational prediction methods. In this chapter, we conduct a comprehensive review of recent computational methods, especially machine learning-based methods, for protein fold recognition. The characteristics of the protein fold recognition problem are described from a computational point of view.

2.1 Characteristics of Protein Folding Problem

The protein folding problem is the question of how protein's amino acids fold into a unique three-dimensional conformation. The first emergence of the protein folding notion was around 1960, with the appearance of the first atomic-resolution protein structures. The firstly discovered protein structures have helices that are packed together in unexpected irregular ways. However, some form of internal crystalline regularity has been previously estimated [5, 8, 13], and α-helices

have been anticipated [5, 8, 14]. Since then, the protein folding problem has been regarded as three different problems:

- **The folding code**: for a given amino acid sequence, what balance of interatomic forces dictates the structure of the protein (thermodynamic)?
- **Protein structure prediction**: how to predict a protein's native structure from its amino acid sequence (computational)?
- **The folding process**: what routes or pathways some proteins use to fold so quickly (Kinetics)?

A variety of factors are identified to determine the probable folding scenarios [13, 15, 14]. Many of the distinct folding mechanisms that emerge depend on the temperatures, which determine the phases of the amino acid chain [14]. Such findings explicitly link the underlying thermodynamic properties of proteins and their folding mechanisms. Several studies focus on the factors that determine the folding rates of two-state proteins. Probable relationships between folding rates and the contact order [14], which emphasize the role of structures involving proximal residues, stability, and Z-score, are established.

Several computational and phenomenological approaches are employed to find the general principles that control the folding rates and mechanisms of single-domain globular proteins [14]. It may be naively thought that the computational protocol for describing protein folding is

straightforward. Because Newton equations of motion fully describe the folding dynamics, and folding may be directly monitored from an appropriately long trajectory. However, there are two severe limitations that prevent this approach from studying protein folding. First, the force fields for such a complex system are not precisely known. As a result, one needs to rely on the transferability hypothesis that interactions derived for small molecules can be used in larger systems, such as proteins. The second problem is simple: the limitations of current computation power. Repeated folding of even a single-domain protein requires the generation of multiple trajectories on a millisecond timescale. Even the creative use of massively parallel computing systems does not entirely address the simulation problem under this severe numerical constraint [15, 14].

Due to these challenges, machine learning approaches for protein fold recognition take the central stage since the emergence of the work described in [21]. Many methods are developed, which are used to assign folds to protein sequences. Machine learning-based methods for protein fold recognition assume [21] that the number of protein folds in the universe is limited, and therefore the protein fold recognition can be viewed as a fold classification problem: using sequence-derived features of proteins whose structure is known, so-called the learning or training set for the construction of a classifier that can then be used to assign a structure-based label to an unknown protein. The procedure of constructing a classifier is called supervised learning or

classifier training. Its role in the fold classification task is to induce a mapping from primary sequences to folding classes.

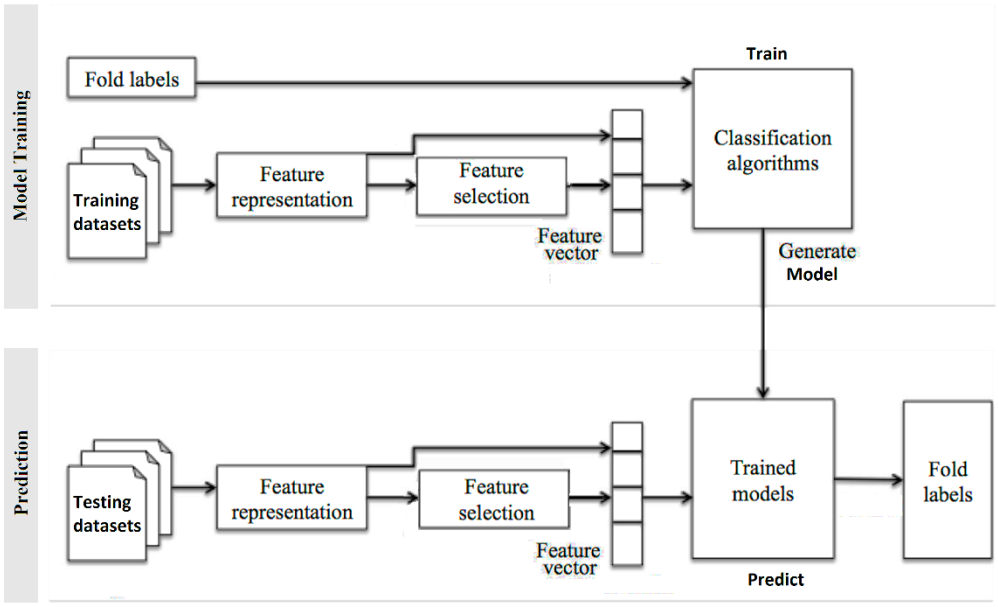


Fig. 12. Supervised Machine learning model for fold recognition. Reproduced from [21].

Fig. *12* shows that the overall procedure in protein fold recognition by machine learning-based methods include two phases: (1) model training and (2) prediction. In the first phase, model training, target protein sequences are first submitted into a feature representation model, in which sequences of different lengths are encoded with fixed-length. The algorithms often used in fold recognition model building include Artificial Neural Network (ANN), Deep Learning (DL), Support Vector Machine (SVM), Random Forest (RF), Naïve Bayes (NB), and Logistic Regression (LR).

In the second phase (prediction), uncharacterized target proteins are submitted into the same feature representation model as in the first phase. Finally, the resulting feature vectors are fed into the trained prediction model, wherein the protein fold class to which the query proteins belong is predicted.

2.1.1 Deep Neural Networks

The basic structure of Deep Neural Networks (DNN) consists of an input layer, multiple hidden layers, and an output layer, as shown in Fig. *13*. After the input data are given to the DNN, the output values are computed sequentially along the layers of the network. The input vector at each layer, comprising the output values of each unit in the layer below, is multiplied by the weight vector for each unit in the current layer to produce the weighted sum [22]. Then, a nonlinear function, such as a sigmoid, hyperbolic tangent, or rectified linear unit (ReLU) [23], is applied to the weighted sum to compute the output values of the layer. The computation in each layer transforms the representations in the layer below into slightly more abstract representations [22, 23]. Based on the types of layers used in the DNN and the corresponding learning method, DNN can be classified as Multi-Layer Perceptron (MLP), Stacked Auto-Encoder (SAE), or Deep Belief Network (DBN).

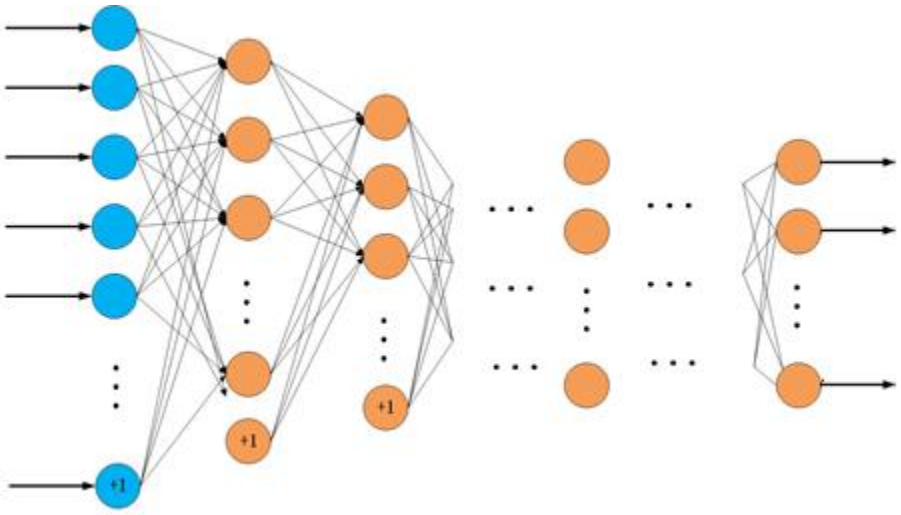


Fig. 13. Deep Neural Network basic architecture.

MLP structure is similar to the usual neural network structure, but includes more stacked layers. It is a purely supervised training system that uses only labeled data. Since the training method is a process of optimization in high-dimensional parameter space, MLP is typically used when a large number of labeled data are available [22, 23].

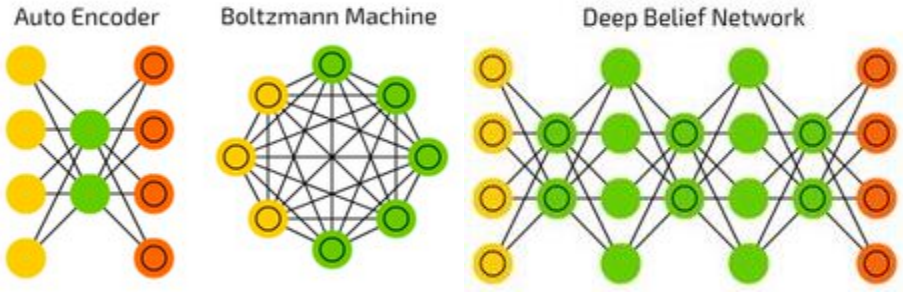


Fig. 14. Different architecture of deep neural network

SAE and DBN use Auto-Encoders (AE) and Restricted Boltzmann Machine (RBM) as building blocks of the architectures, respectively. The main difference between these and MLP is that training is executed in two phases: unsupervised pre-training and supervised fine-tuning. First, in unsupervised pre-training (Fig. *14*), the layers are stacked sequentially and trained in a layer-wise manner as an AE or RBM using unlabeled data. Afterwards, in supervised fine-tuning, an output classifier layer is stacked, and the whole neural network is optimized by retraining with labeled data. Since both SAE and DBN exploit unlabeled data and can help avoid overfitting, researchers are able to obtain regularized results, even when labeled data are insufficient, which is a common situation in the real world [23].

DNNs, as hierarchical representation learning methods, can discover previously unknown highly abstract patterns and correlations to better understand the nature of the data. However, the capabilities of DNNs have not yet fully been exploited. Although the key characteristic of DNNs is that hierarchical features are learned solely from data, human-designed features are given as inputs instead of raw data forms. The progress of DNNs comes from investigations into proper ways to encode raw data and learn suitable features from them.

2.1.1.1 Convolutional Neural Networks Architectures

Convolutional Neural Networks (CNNs) are directly inspired by the visual cortex of the brain. In the visual cortex, there is a hierarchy of two basic cell types: simple cells and complex

cells [23]. Simple cells react to primitive patterns in sub-regions of visual stimuli, and complex cells synthesize the information from simple cells to identify more intricate forms. Hence, CNNs are applied to imitate three key ideas: local connectivity, invariance to location, and invariance to local transition. The basic structure of CNNs consists of convolution layers, nonlinear layers, and pooling layers, as shown in Fig. *15*. In order to use highly correlated sub-regions of data, feature maps, which are groups of local weighted sums, are obtained at each convolution layer. The feature maps are achieved by computing convolutions between local patches and weight vectors called filters. Furthermore, since identical patterns can appear regardless of the location in the data, filters are applied repeatedly across the entire dataset, which also improves training efficiency by reducing the number of parameters to learn. Then nonlinear layers increase the nonlinear properties of feature maps. At each pooling layer, maximum or average subsampling of non-overlapping regions in feature maps is performed. This non-overlapping subsampling enables CNNs to handle fairly different but semantically similar features and thus aggregate local features to identify more complex features. Currently, CNNs are one of the most successful deep learning architectures owing to their outstanding capacity to analyze spatial information.

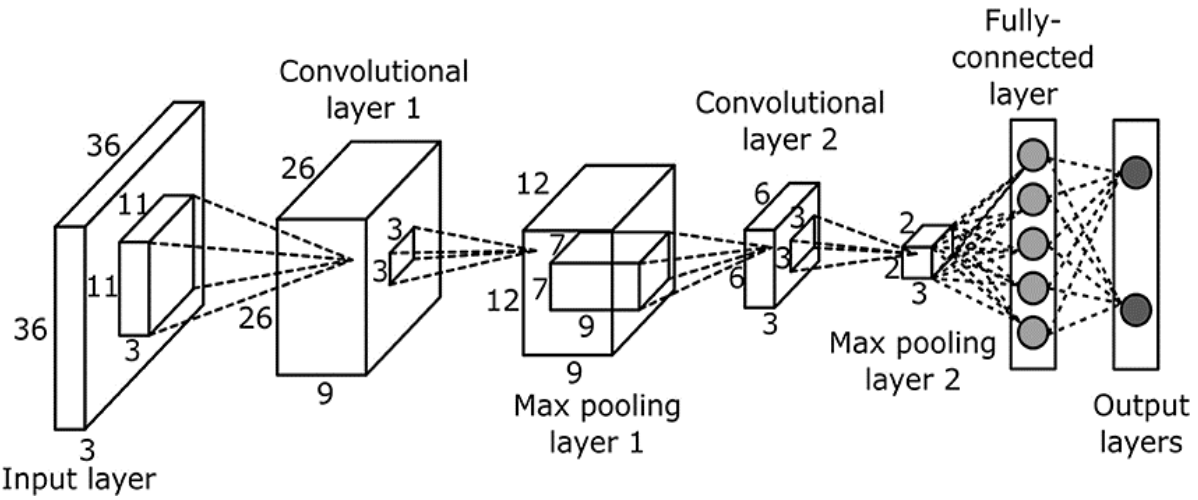


Fig. 15. CNN

Over the years, variants of this fundamental architecture are developed, leading to amazing advances in the field. A good measure of this progress is the error rate in competitions, such as the ILSVRC ImageNet challenge. In this competition LeNet-5 architecture [24], AlexNet [25], GoogLeNet [26], and ResNet [27] contribute to image classification domain, where the top-5 error rate fall from over 26% to barely over 3% in just five years.

2.1.2 Random forest

In [28] the decision tree methods are introduced, it is widely used in many domains due to its simplicity and good interpretability. Conversely, the accuracy of a single decision tree is often lower than more advanced classification methods such as support vector machines or neural networks. The recent developments in the decision tree find that using an ensemble of decision trees, constructed from randomly selected features and training data, often yield to significantly

higher accuracy [29]. This advanced approach is called random forest. Random forest is a meta-learning algorithm for classification, which consists of a bag of separately trained decision trees. Therefore, it inherits the advantages of decision tree methods such as easy training, fast prediction, and good interpretability. In the random forest, the average prediction of the decision trees is robust against the existence of irrelevant features, because it selects a random subset of the input features to construct each decision tree. Furthermore, the random selection of a subset of the training data to train each tree also leads to an ensemble of decision trees that are resistant to noise and disproportional class distribution in the training data. Fig. *16* illustrates how the random forest makes a prediction.

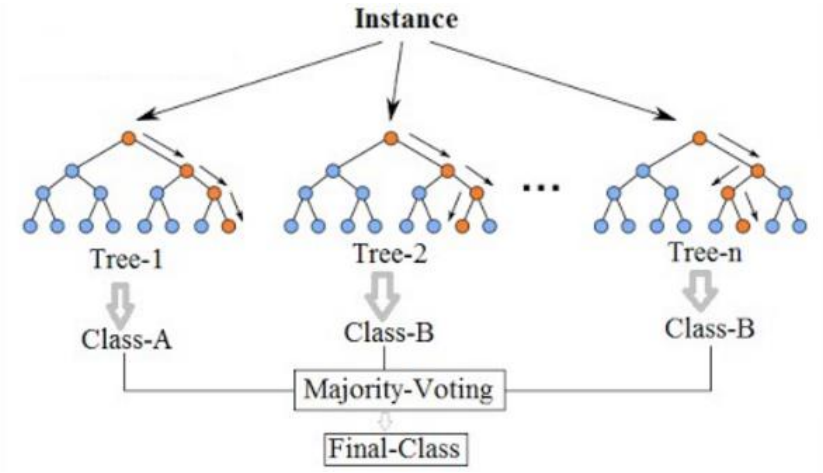


Fig. 16. Random Forest method.

2.2 Protein Fold Recognition Datasets

A public benchmark dataset of protein fold recognition is usually used to examine the effectiveness of existing machine learning-based methods. In the literature of protein fold recognition, there are three popular benchmark datasets: Ding and Dubchak (DD) [14], Taguchi and Gromiha (TG) [30], and Extended-DD (EDD) [31] (see Appendix 1).

DD-dataset, designed by Ding and Dubchak [14], is used in several studies as shown in Table 2. It is comprised of a training dataset and a testing dataset, both of which cover 27 protein folds in the SCOP database, which belong to different structural classes containing α, β, α/β, and α+β, comprehensively. DD's training dataset contains 311 protein sequences with ≤40% residue identity, and the testing dataset contains 383 protein sequences with ≤35% residue identity. Additionally, the sequences in the training dataset have identity ≤35% with that in the testing dataset, thus ensuring to provide an unbiased performance evaluation. The sequence distribution of each of the 27-fold classes can be seen in **Error! Reference source not found.** (Appendix I).

The DD dataset suffers some limitations. For instance, the DD dataset is imbalanced, as shown in Table 8. (Appendix I) the ratio of the smallest class, EF hand-like, against the largest class, immunoglobulin-like β-sandwich is roughly 1:4. Moreover, the sample size is small for each fold class, only 383 training sequences belong to 27-fold classes, the samples in each class range from 6 to 30.

The second benchmark is TG dataset, which contains 1,612 protein sequences belonging to 30 different folds from SCOP version 1.73 constructed by Taguchi and Gromiha [30]. The benchmark with the detailed information of the 30 different fold types is described in [14], and the sequence identity between two proteins is no more than 25%. Table 10 (Appendix I) shows the TG benchmark.

EDD dataset is the third benchmark. EDD contains 3,418 protein sequences, which belong to the 27 different folds that are essentially used in the DD dataset from SCOP version 1.75. EDD has more sequences in each fold than DD, and TG [14], and the sequence identity between the two proteins is no more than 40% (Table 9 Appendix I).

2.3 Framework of Machine Learning-Based Methods for Protein Fold Recognition

One of the most essential tasks in structural bioinformatics is protein fold classification. As protein folding information is helpful in identifying the tertiary structure and functional information of a protein [5]. Recently, many protein fold recognition studies have been developed by means of machine learning. Machine learning-based protein fold recognition methods can be categorized into two classes according to the learning algorithms used: (1) single classifier-based methods; and (2) ensemble classifier-based methods.

2.3.1 Single Classifier-Based Methods

Currently, most of the single classifier methods used in protein fold recognition are based on SVM. Since SVM is a well-known classification algorithm and is highly efficient in several fields of bioinformatics. Some for SVM-based protein fold recognition methods are: [32], ACCFold_AC and ACCFold_ACC [31], TAXFOLD [33], and Alok Sharma's method [34]. The main difference between these methods is their feature representation methods. For instance, [32] uses secondary structural state and solvent accessibility state frequencies of amino acids and amino acid pairs as feature vectors. Hence, among these features, the secondary structural state frequencies are the most effective features for fold class discrimination. However, combining the secondary structural state frequencies with the other two features can further improve the accuracy of fold discrimination.

In ACCFold_AC and ACCFold_ACC methods, the features are based on the distant evolutionary relationships of protein sequences, which can effectively capture the evolutionary information embedded in the form of Position-Specific Score Matrices (PSSM) [35]. The TAXFOLD [33], suggests using global and local sequential and structural features for protein fold classification. Given that an increase in the number of features is probably not an informative mean to further improve recognition accuracy. Thus, a classification method that can assess the contribution of these potentially heterogeneous object descriptors must be developed. Therefore,

[36] proposes a single multi-class kernel machine that informatively combines available feature groups.

In addition to SVM classifier, other single classifiers, such as Random Forest [37] and HMM [38], are used to construct a prediction engine for protein fold recognition methods. For instance, [37] proposes an RF-based protein fold recognition method called PFP-RFSM. The structure of PFP-RFSM involves a comprehensive feature representation algorithm that can capture distinctive sequential information from the protein sequence and structural information from predicted structures. These features have seven perspectives, namely: amino acid composition, secondary structure contents, predicted relative solvent accessibility, predicted dihedral angles, PSSM matrix, nearest neighbor sequences, and sequence motifs. PFP-RFSM is the first protein fold recognition method to utilize features based on sequence motifs. Furthermore, the PFP-RFSM method is the first to use the RF classifier as its prediction engine. RF classifier is superior over the other commonly used classifiers in the overall performance. Alternatively, [38] proposed an optimization method for protein fold classification; the prediction model of this method is constructed based on a Markov chain trained on the primary structure of proteins. Additionally, the presented model is tested on a reduced state-space HMM, which is an effective means of classifying proteins in fold categories with low computational cost.

2.3.2 Ensemble Classifier-Based Methods

The most recently developed methods for protein fold recognition are based on ensemble classifier models. In [39], a popular ensemble classifier method is presented (PFP-FunDSeqE), which has a new feature extraction method to explore the functional domain information and sequential evolution information. This method generates 17,402 functional domain features and 220 Pseudo PSSM features. The two feature groups are separately fed into an optimized evidence-theoretic K-Nearest Neighbor OET-KNN classifier to build prediction models.

Moreover, a protein fold recognition method called PFPA is presented in [40]. PFPA employs a novel feature representation algorithm that considers the sequential evolutionary information and structural information. The sequential evolutionary information is resulting from PSI-BLAST [35] profiles which are produced by searching query proteins against a non-redundancy database. Based on the PSI-BLAST profiles, PFPA computes 20 PSSM features and 420 amino acid compositional features from consensus sequences, which contain rich evolutionary information. The structural information is resulting from PSI-PRED [41] profiles. To sufficiently explore the structural information, PFPA calculates 27 local and 6 global secondary structure features from PSI-PRED profiles. Regularly, an integration of all the sequential and structural features is developed to construct comprehensive feature representations of target proteins. For the prediction engine, an ensemble classifier model is constructed, which makes use of five basic

classifier models RF, NB, Bayes Net, LibSVM, and Sequential Minimal Optimization SMO with an average probability strategy.

Recently, [42] has developed a recognition method called ProFold. ProFold initially considers using protein tertiary structure information in its feature extraction framework. Successively, other commonly used features, such as global features of amino acid sequence, PSSM features, functional domain features, and physiochemical features are used. The tertiary structure features are employed to compute eight types of secondary structure states from PDB files by using DSSP. ProFold proposes a novel strategy to construct an ensemble classifier. Primarily, the paper selects 10 widely used basic classifiers, such as Logistic model tree [43], RF, LibSVM, Simple Logistic, Rotation Forest, SMO, NB, Random Tree, Functional tree, and Simple Cart. Subsequently, distinct types of feature representations are trained using these 10 basic classifiers. For each feature type, the model with the highest accuracy is chosen, generating four single classifier models for the four feature types. These models are DSSP model, AAsCPP model, PSSM model, and functional domain model. The average probability strategy is used to fuse the four single classifier models, similar to that in the PFPA method.

Table 2 lists the evaluation of twenty methods published in the past twelve years, from 2006 to present, on the DD dataset. From the evaluation, we observe the following:

- ProFold shows the best performance among other methods. The overall accuracy of ProFold is 76.2%, which is 2.6%–15.7% higher than the other methods. It is illustrated that the ProFold has great power to distinguish the 27-fold classes in the DD dataset. This significant enhancement of ProFold is due to the use of the DSSP features. These results indicate that integrating the DSSP features into feature representations is a remarkable enhancement [44].
- Fourteen methods are based on an ensemble classifier, while six methods are based on a single classifier.
- Nine methods that obtain an overall accuracy >70% are PFP-FunDSeqE 70.5%, TAXFOLD 71.5%, Marfold 71.7%, Kavousi et al. 73.1%, PFPA 73.6%, Feng and Hu 70.2%, Feng et al. 70.8%, and ProFold 76.2%, respectively. Notice that TAXFOLD is the only method that is based on a single classifier while the other methods are based on ensemble classifier.

The results in Table 2 indicate that ensemble classifiers are more effective than single classifiers for protein fold recognition. They demonstrate accurate, robust, and reliable performance. Also, they can be applied in large-scale protein fold recognition. They can effectively address the intrinsic limitations of experimental methods, that is, being time consuming and expensive.

Table 2

Top 20 protein fold recognition methods results on DD datasets.

Index	Methods	Year	Classifier Type	Overall Accuracy (%)
1	Nanni et al. [45]	2006	Ensemble	61.1
2	PFP-Pred [46]	2006	Ensemble	62.1
3	Shamim et al. [32]	2007	Single	60.5
4	PFRES [47]	2007	Ensemble	68.4
5	Damoulas et al. [36]	2008	Single	68.1
6	ALHK [48]	2008	Ensemble	61.8
7	GAOEC [49]	2008	Ensemble	64.7
8	PFP-FunDSeqE [39]	2009	Ensemble	70.5
9	ACCFold_AC [31]	2009	Single	65.3
10	ACCFold_ACC [31]	2009	Single	66.6
11	Ghanty et al. [50]	2009	Ensemble	68.6
12	TAXFOLD [33]	2011	Single	71.5
13	Alok Sharma et al. [34]	2012	Single	69.5
14	Marfold [51]	2012	Ensemble	71.7
15	Kavousi et al. [52]	2012	Ensemble	73.1
16	PFP-RFSM [37]	2013	Single	73.7
17	Feng and Hu [53]	2014	Ensemble	70.2
18	PFPA [40]	2015	Ensemble	73.6
19	Feng et al. [54]	2016	Ensemble	70.8
20	ProFold [42]	2016	Ensemble	76.2

CHAPTER III

DECODING THE STRUCTURAL KEYWORDS IN PROTEIN STRUCTURE UNIVERSE

Protein fragments are widely used in a varied range of applications, such as comparing protein structures through reduced representations of fragments, modeling homologs at the fragment level, investigating sequence-structure relationships, approximating tertiary structures, modeling loop conformations, and predicting novel folds. The quality of the fragment libraries plays a critical role in these structural biology applications.

The continuously increasing number of high-resolution, experimentally determined protein structures provides rich protein structure sources that enable us to generate high-quality fragment libraries. Moreover, regarding the length of the appropriate fragments, Handl et al. [55] report that the longer the fragments are, the more useful they are in structure prediction. The increasing number of experimentally determined protein structures also enables us to derive libraries of longer fragments and then use them together with the short ones to form a rich fragment dictionary to decode the protein structure universe. Usually, protein fragment libraries are constructed based on clustering similar protein backbone conformations.

In this chapter, we present a generated large-scale protein fragment sample sets, called Frag-K, with lengths ranging from 4 to 20 residues. Frag-K is developed from a large number of non-homogenous protein structures covering diverse conformations in the protein structure universe. To generate Frag-K, we apply a spectral clustering algorithm to aggregate these fragment samples according to their structural similarity. A rank-revealing randomized singular value decomposition (R^3SVD) algorithm [56] is employed to fast approximate the dominant eigenvectors of the fragment affinity matrices, which enables the spectral clustering method to scale up to large fragment sample sets. The representative fragment in each cluster is then collected to assemble the fragment library. Moreover, with fragment sample sets of significantly larger sizes, we are able to generate long protein backbone fragment libraries up to 20 residues. We further identify the most sensitive clustering cut-off values with respect to fragment libraries of different lengths in distinguishing protein folds. Finally, these fragments are collected as a structural dictionary to train a random forest to classify protein structures in popular SCOP folds. Our feature selection results show that a structural dictionary with ~400 fragments of different lengths is capable of classifying major SCOP folds with high accuracy and fragments of different lengths contribute.

3.1 Methodology

3.1.1 Generation of Fragment Libraries

By applying randomized spectral clustering, iterative bi-partitioning, and random forest classifier, we generate 4- to 20-residue fragment libraries that can be effectively encoded as structural features in distinguishing between protein folds. First of all, for all fragment samples of the same length, we construct a fragment affinity graph whose edges are weighted by the pairwise C_α Root Mean Square Deviation (RMSD) between every two fragments. Then, a randomized spectral clustering algorithm is applied to the affinity matrix corresponding to the weighted fragment graph to approximate the dominant eigenvector to bi-partition the graph into two complementary sub-graphs. The bi-partitioning process is repeated on the subsequent subgraphs until the pairwise C_α RMSD among all fragments in the subgraphs is within a pre-specified cutoff value. All fragments in each subgraph form a cluster sharing structural similarity. The fragment having most similar fragments in the cluster given the clustering cutoff is exacted as a representative fragment of the cluster and is then deposited into the Frag-K fragment libraries. The small clusters with less than 3 fragments are ignored. By specifying RMSD cutoffs from $0.1\dot{A}$ to $4.0\dot{A}$ with $0.1\dot{A}$ increment, we generate fragment libraries with respect to different clustering cutoffs. Afterward, for each fragment library of a certain length, we encode all fragments in the fragment library into a structural feature vector and then apply a random forest classifier to classify the SCOP-40 proteins into four major protein structure classes (all-α, all-β, α/β, and α+β).

According to the performance of the fragment library with different clustering cutoffs in classifying SCOP-40 proteins, we identify the most appropriate RMSD cutoffs for each fragment length. Figure 17 illustrates the overall flowchart of generating Frag-K libraries.

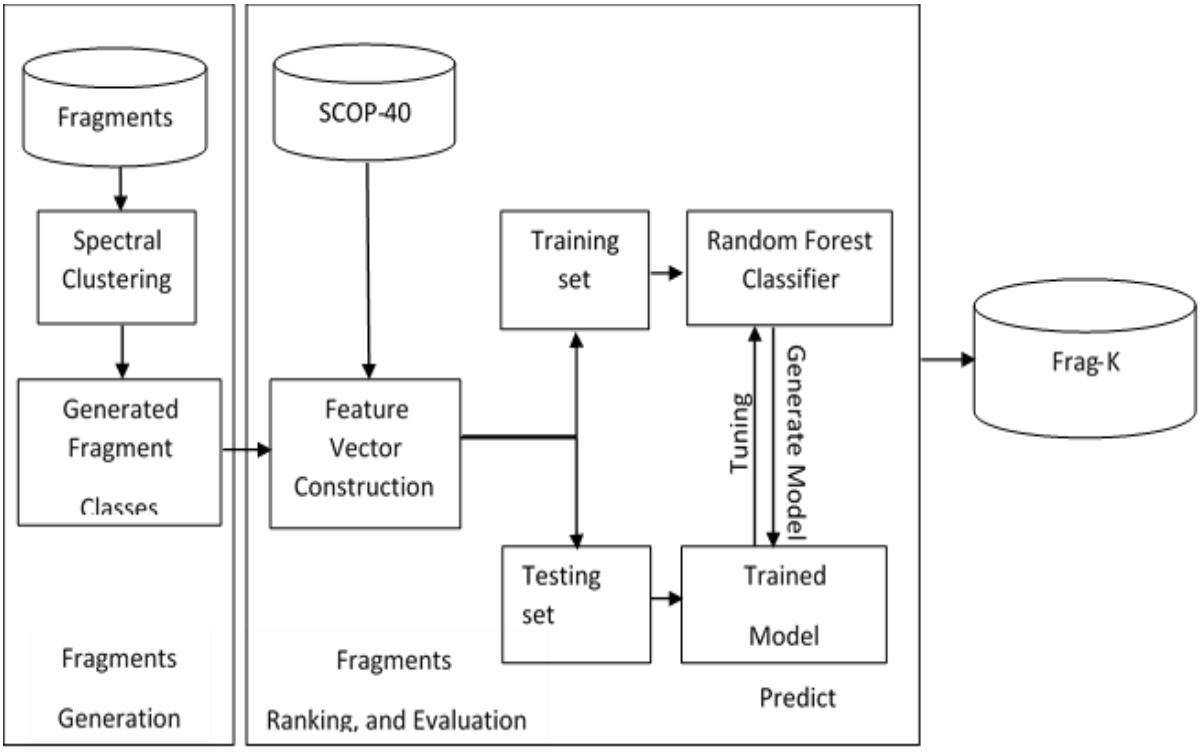


Fig. 17. Generation of Frag-K Libraries.

3.1.2 Fragment Affinity Matrices

Given a pair of fragments f_i and f_j of the same length, we superimpose them to minimize the C_α atom deviations between the fragment pair then calculate the RMSD values of the corresponding C_α atoms, which gives the distance score between these two fragments. An undirected, weighted fragment affinity graph $G = (V, E, a)$ is created where $f_i \in V$ and $(f_i, f_j) \in$

E if the RMSD value between fragments f_i and f_j is within the pre-specified RMSD cut-off τ. The corresponding connection affinity $a(f_i, f_j)$ is calculated by applying the Gaussian kernel to convert the RMSD value to the affinity score such that

$$a(f_i, f_j) = \begin{cases} \exp(-\frac{rmsd(f_i, f_j)}{\sigma^2}) & rmsd(f_i, f_j) \leq \tau \\ 0 & rmsd(f_i, f_j) > \tau \end{cases},$$

where σ^2 is the overall standard deviation of the RMSD distribution of the fragment sample set. Then, a fragment affinity matrix A corresponding to G is generated, where $A_{ij} = a(f_i, f_j)$. Due to the nonnegative property of the Gaussian kernel and the commutative property of RMSD, A is Symmetric Positive Definite (SPD). Moreover, A is sparse when an efficient RMSD cutoff is applied.

3.1.3 Randomized Spectral Clustering

Randomized spectral clustering is a scalable spectral clustering method designed to reduce the computation cost operation of calculating the bi-partitioning eigenvectors of the large affinity matrix. Unlike the classical clustering techniques such as the k-means approaches, the spectral clustering method is able to produce clusters with concave cluster boundaries due to the nonlinear separation hyper-surfaces obtained. As a result, spectral clustering does not need any prior information on the shapes of the clusters. Moreover, if the bi-partitioning eigenvectors are computed accurately, spectral clustering yields more robust clustering results because it does not rely on the initial, randomly selected cluster centers.

Spectral clustering [57] is a graph-based clustering technique [58] that can be viewed as finding the bi-partitions of a graph by minimizing the graph cut property. The fundamental idea of spectral clustering [59] is to make use of the spectrum (eigenvalues/eigenvectors) of the affinity matrix with respect to a graph $G = (V, E, a)$ to perform dimensionality reduction before clustering in lower dimensions. Starting from the fragment affinity matrix A of G, a diagonal matrix D is defined as $D_{ii} = \sum_{j=1}^{n} A_{ij}$. Then, a normalized Laplacian matrix L is constructed such that $L = D^{-1/2}AD^{-1/2}$. Given two complementary partitions S and $\bar{S}$ such that $S, \bar{S} \subseteq V$, $S + \bar{S} = V$, and $S \cap \bar{S} = \emptyset$, the normalized cut property $ncut(S, \bar{S})$ is defined as

$$ncut(S, \bar{S}) = \frac{w(S, \bar{S})}{w(S, V)} + \frac{w(S, \bar{S})}{w(\bar{S}, V)}$$

where $w(X, Y)$ is the weight function summing all pairwise weights between vertices in X and those in Y. Hence, $ncut(S, \bar{S})$ measures the balanced similarity between S and $\bar{S}$. According to the theory of spectral clustering, the eigenvector corresponding to the largest eigenvalue of L forms a graph cut that minimizes $ncut(S, \bar{S})$. Therefore, we calculate the eigenvector with respect to the largest eigenvalue of the Laplacian matrix L generated from the fragment affinity matrix A to bi-partition fragments of the same length. The bi-partitioning process is repeated until the pairwise distance among the fragments in the partition is less than the pre-specified RMSD cut-off value τ.

The most computationally costly operation in the spectral clustering method is the calculation of the bi-partitioning eigenvector from the Laplacian matrix L to bi-partition G as well

as its subsequent sub-graphs, particularly when a large number of fragment samples are involved. Fortunately, we only need the dominant eigenvector and thus there is no need to calculate the whole spectrum of L. Moreover, because the normalized Laplacian matrix L is SPD, its eigenvalue decomposition and singular value decomposition (SVD) coincide. Therefore, we adopt a rank-revealing randomized singular value decomposition (R^3SVD) algorithm [56] to fast approximate the dominant eigenvector of the normalized Laplace L matrix.

The R^3SVD algorithm includes four major steps: Gaussian sampling, QB decomposition, error estimation, and SVD. First of all, in Gaussian sampling, given an $n \times n$ Laplacian matrix L, an $n \times k$ Gaussian matrix Ω is randomly generated and an $n \times k$ matrix Y is obtained by projecting L onto Ω such that $Y = L^q\Omega$ using power iteration, where $k \ll n$ is the guessed rank and q is the number of power iterations. Here, we adopt $q = 2$ as recommended by [60]. Then, a QB decomposition is carried out, where Q is generated by a QR decomposition on Y such that $[Q, R] = qr(Y)$ and B is obtained by projecting Q^T onto L such that $B = Q^T L$. Consequently, $QB \approx L$ is a k-rank approximation of L. The relative error of the QB decomposition can be efficiently computed by calculating the squares of the Frobenius norms of L and B such that

$$\frac{\|L - QB\|_F^2}{\|L\|_F^2} = \frac{\|L\|_F^2 - \|B\|_F^2}{\|L\|_F^2}.$$

The mathematical proof of the above property can be found in [61]. Due to the assumption that there is a limited number of independent factors that determine the formations of structures of

short protein fragments, the Laplacian matrix L is of low rank. As shown in [62], if the relative error of the QB decomposition is sufficiently small, the dominant eigenvector of L can be approximated with high precision. R^{3}SVD employs an adaptive way by repeating the Gaussian sampling step with a gradually increasing rank k to control the relative error of the QB decomposition below the desired threshold. Afterward, the low-rank approximated SVD of $L, U_L\Sigma_L V_L^T$, is obtained by carrying out SVD on the "short-and-wide" matrix B such that $[U_B, \Sigma_B, V_B] = svd(B)$. Then, $U_L = Q^T U_B$, $\Sigma_L = \Sigma_B$, and $V_L = V_B$. $U_L\Sigma_L V_L$ is a low-rank approximation of L. Finally, the approximated dominant eigenvector of L can be extracted from U_L. The R^{3}SVD algorithm is able to adaptively estimate the appropriate rank of the approximated $U_L\Sigma_L V_L^T$ to calculate the dominant eigenvector of L. In the randomized algorithm, most numerical linear algebraic operations are carried out on "tall-and-skinny" block matrices, which are both efficient in computation and memory. This allows the spectral clustering method to scale up to handle the large datasets in this study with close to half a million protein fragments.

3.1.4 Finding the Optimal RMSD Cutoffs

We use the randomized spectral clustering algorithm to generate a series of fragment libraries subject to RMSD cutoffs from 0.1 Å to 4.0 Å with 0.1 Å increment. In fact, these fragment libraries are sensitive to the RMSD cutoff values in the randomized spectral clustering algorithm. If the RMSD cutoff is too small, there may be too many highly structurally similar clusters. On

the other hand, if the RMSD cutoff is too big, some important fragments may be missed due to being included into another cluster represented by the other fragments during the clustering process. Moreover, the most appropriate RMSD cutoffs for fragment libraries of different lengths are likely to be different, which need to be carefully justified.

Here, we employ the SCOP-40 dataset to measure the performance of the generated fragment libraries with respect to different RMSD cutoffs as structural features to classify protein structures into four major protein structure classes (all-α, all-β, α/β, and α+β) so as to identify the most appropriate clustering RMSD cutoffs for fragment libraries of different lengths. We use a "bag-of-words" model to represent a protein structure as a structural feature vector. More precisely, given a fragment library of length l, a fragment feature vector is formulated as $F = [f_1, f_2, \dots, f_n]^T$, where f_i is the frequency of the ith fragment in the fragment library and n is the size of the fragment library. Then, we use a sliding window of length l to consecutively segment a protein structure into overlapping l-residue fragments. Gaps are excluded. If the pairwise RMSD of a fragment in the protein structure to a fragment in the fragment library is within the RMSD cutoff threshold, it is regarded as a match. As a result, a protein structure is encoded as a fragment vector.

A random forests classifier based on growing unbiased trees [63], which can effectively avoid the uncertainty of feature rankings, is trained to classify the protein structures in SCOP-40

into four major protein structure classes (all-α, all-β, α/β, and α+β). The random forest training process is carried out on each fragment library with respect to different lengths and RMSD cutoffs. Then, the fragments in the fragment library are ranked according to the impurity decrease in the random forest and the RMSD cutoffs in generating the fragment libraries are justified according to the testing results.

We randomly select 70% of the protein structures in each structure class in the SCOP-40 dataset to construct a training set and the rest 30% forms a test set. The training set is used to train the random forest classifiers via 10-fold cross validation for fragment libraries of different lengths and generated with different RMSD cutoffs. Figure 18 shows the accuracies of the random forest classifiers on the test set using our 4-, 12-, and 20-residue fragment libraries generated with RMSD cutoffs ranging from 0.1Å to 4.0Å. One can find that the optimal accuracy occurs at RMSD cutoffs of 0.4 Å, 1.3Å, and 2.2Å for fragment libraries of lengths of 4, 12, and 20 residues, respectively. In a word, the clustering RMSD cutoff plays an important role in the performance of the generated fragment library as structural features. Moreover, Table 3 lists the RMSD cutoffs for fragment libraries of lengths ranging from 4 to 20 residues as well as the total number of fragments with the optimal capability to be encoded as structural features to distinguish among protein folds. Without surprise, the optimal RMSD cutoffs increase nearly proportionally with fragment lengths. Moreover, it is interesting to notice that the sizes of the fragment libraries do not increase either

monotonically or proportionally. For example, the number of 8-residue fragments is over three times more than that of the 7-residue ones. This is due to the fact that a significant portion of 7-residue fragments forms α-helices, which result in a smaller number of clusters. Moreover, the numbers of fragments in fragment libraries over 13 residues start to decrease with length. This is because the longer fragments are more structurally diversified, which results in a lot of small clusters with the fragments below the specified threshold.

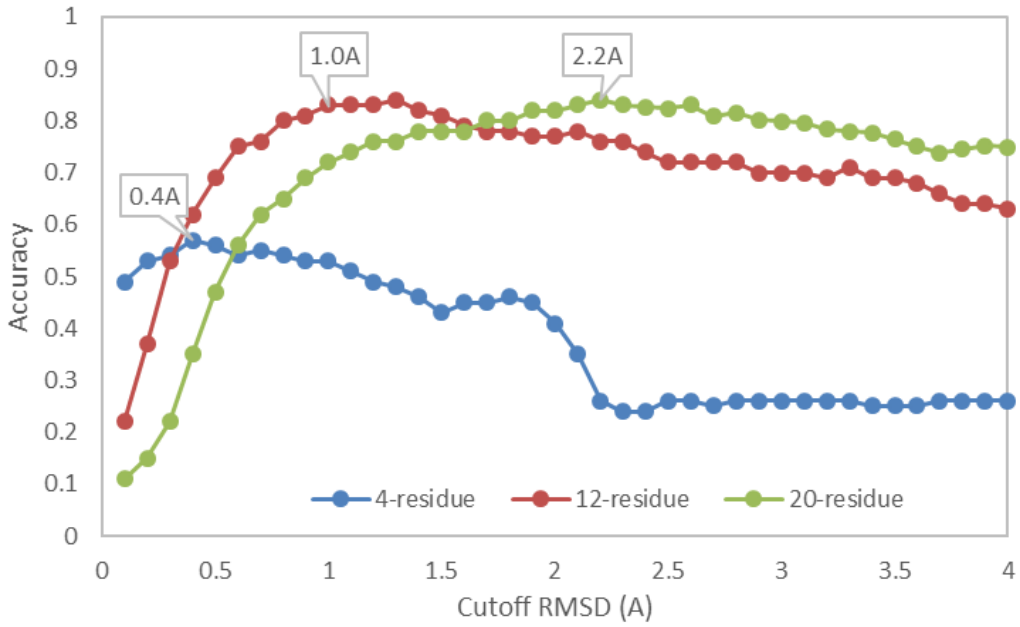


Fig. 18. Comparison of classification accuracies of major protein structure classes (all-α, all-β, α/β, and α+β) on SCOP-40 proteins using 4-, 12-, and 20-residue fragments as structural features. The performance of the fragment libraries is sensitive to the RMSD cutoffs. The optimal RMSD cutoffs for 4-, 12-, and 20-residue fragment libraries are 0.4A, 1.0A, and 2.2A, respectively.

Table 3

The optimal RMSD cutoffs and the number of fragments for Frag-K of different lengths

Length	Optimal RMSD Cutoff (A)	# of Fragments
4	0.4	496
5	0.6	1145
6	0.7	682
7	0.7	1250
8	0.7	4050
9	0.7	4500
10	0.8	7745
11	1	7945
12	1	7370
13	1	7434
14	1.1	6947
15	1.2	5414
16	1.3	6153
17	1.4	4425
18	1.6	4202
19	1.9	4154
20	2.2	4012

3.2 Datasets

3.2.1 Fragment Sets

We use the Protein Sequence Culling Server (PISCES) [64] to extract a non-redundant and non-homologous set (Cull20) of protein chains from PDB. Cull20 contains 2,491 protein chains with at most 20% sequence identity, 1.6 Å resolution cut-off, and 0.25 R-factor. For each protein chain in Cull20, a fixed-length sliding window is used to consecutively segment the protein sequence into overlapping fragments. Fragments with gaps are excluded. We repeatedly use sliding windows with sizes ranging from 4 to 20 residues to generate 4- to 20-residue fragment

samples, respectively. A reduced fragment representation is employed such that each residue in a fragment sample is encoded by the spatial coordinates of C_α atoms while the other backbone atoms and side chains are removed. Residue identities in each fragment are also ignored. Table 4 lists the total numbers of generated protein fragment samples of different lengths from protein chains in Cull20.

Table 4 Total numbers of fragment samples with respect to fragment lengths in Cull20.

Length	# of Fragments
4	503,252
5	498,792
6	494,382
7	490,044
8	485,766
9	481,540
10	477,375
11	473,266
12	469,210
13	465,188
14	461,217
15	457,295
16	453,421
17	449,583
18	445,785
19	442,018
20	438,295

3.2.2 Testing and Validation Datasets

We use the EDD [31] dataset to train random forests to classify protein structures belonging to different folds where the generated fragment libraries are used as structural features. As shown

in Table 9 (Appendix I), the 27 fold classes in EDD cover most of the SCOP database structures. The EDD dataset is used to compare Frag-K with similar studies in the literature.

The effectiveness of using the fragment libraries as structural features to distinguish between protein folds is sensitive to the RMSD cutoffs used to generate the fragment clusters. We herein construct a SCOP-40 dataset to analyze the impact of the clustering cutoffs on the performance of Frag-K. SCOP-40 is a dataset that hosts proteins with less than 40% sequence identity extracted from SCOPe v2.07 [2]. It contains four major protein structure classes (all-α, all-β, α/β, and α+β) covering approximately 90% of SCOPe v2.07. We use SCOP-40 to build training and test sets to justify Frag-K fragment libraries generated with different RMSD cutoffs in classifying all-α, all-β, α/β, and α+β structure classes. All proteins that belong to EDD dataset are excluded from SCOP-40.

3.2.3 Performance Measures

Fold classification is conducted on the EDD dataset to measure the effectiveness of protein structure classification using fragment libraries as structural features. The classification performance is measured in terms of precision, recall, F-measure, and accuracy such that

$$recall = \frac{TP}{TP + FN},$$

$$precision = \frac{TP}{TP + FP},$$

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall},$$

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN},$$

where TP, TN, FP, and FN denote the numbers of true positive, true negatives, false positive, and false negatives, respectively.

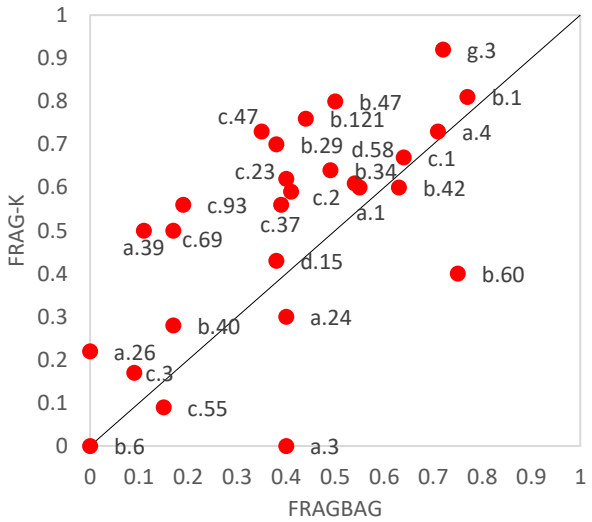
3.3 Results

3.3.1 Analysis of Fixed-length Fragment Libraries

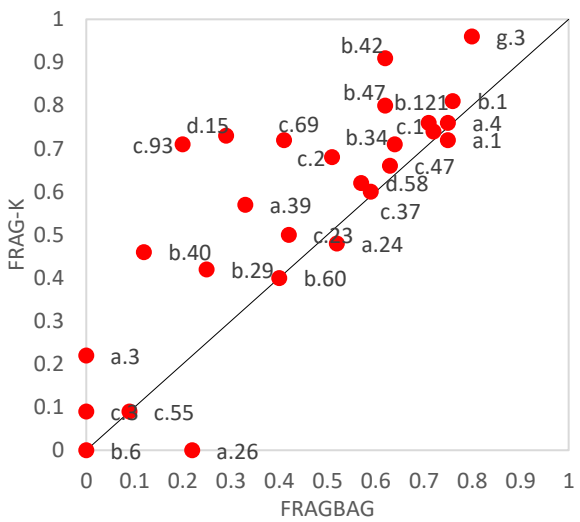
We compare Frag-K with Fragbag developed by Kolodny et al. [65] in their capabilities of distinguishing major protein structural folds in the EDD dataset. To ensure that the test set contains samples from all folds, 30% of protein structures in each fold are randomly selected to form the test set while the rest become the training set. Then, for a fragment library of each length, we train a random forest classifier using Frag-K to encode each protein structure. Similar classifiers are constructed using Fragbag libraries. Figure 19 compares the performance of Frag-K and Fragbag of lengths ranging from 4 to 12 residues, where the X and Y coordinates of each subfigure are the classification accuracies of using Fragbag and Frag-K, respectively. In protein fold classification using short fragments, Frag-K outscores Fragbag in 22, 24, 22, and 25 fold classes out of 27 in 4-, 5-, 6-, and 7-residue fragments, respectively. The advantage of Frag-K widens for longer fragments. In particular, for 12-residue fragments, the classification accuracies of our library are higher than those of Fragbag in almost all fold classes. This is due to the fact that Frag-Ks libraries are effectively derived from many more protein structures available in PDB today than 15 years

ago when Kolodny et al. generated Fragbag, which better represent the structural feature space, particularly for long fragments, in the protein structure universe.

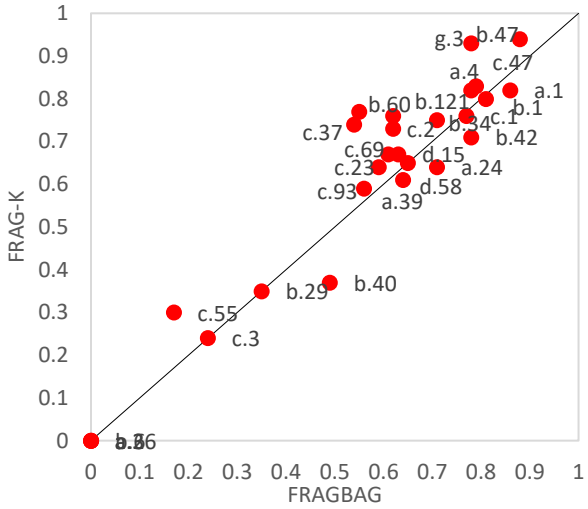
It is important to notice that the longer fragments tend to exhibit better classification capability. Moreover, the α/β folds often yield higher classification accuracy. This is because these longer fragments often capture long segments of secondary structures as well as super-secondary structures [66] such as β-hairpins, short β-sheets, helix-loop-helix, helix-turn-helix, etc., which effectively represent the structural traits of each protein fold. However, for certain folds, shorter fragments seem to be more effective. For example, using 11-residue Frag-K as structural features completely misclassifies a.3 and a.26; however, 4- and 5-residue fragments in Frag-K demonstrate certain success. This indicates that a structural dictionary consisting of fragments of different lengths is likely to demonstrate better classification capabilities than the one with fragments of the same length.



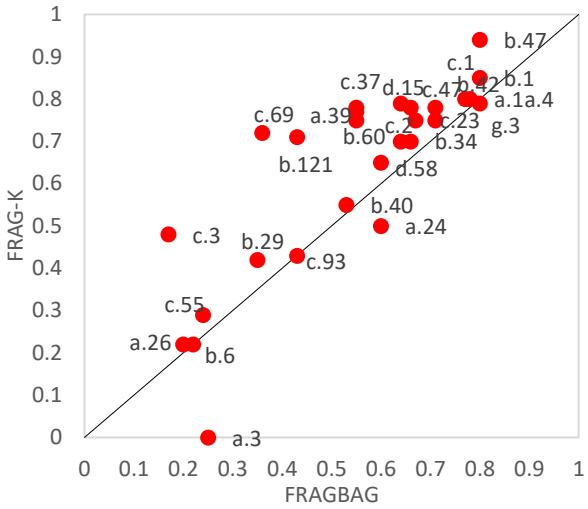
(a). 4-residue fragments



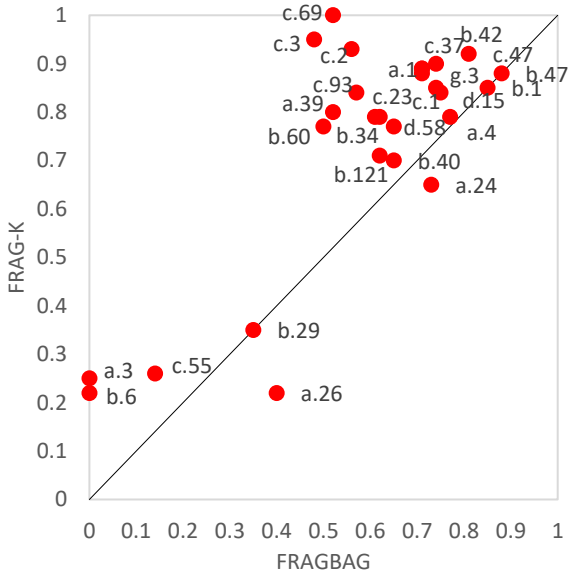
(b). 5-residue fragments



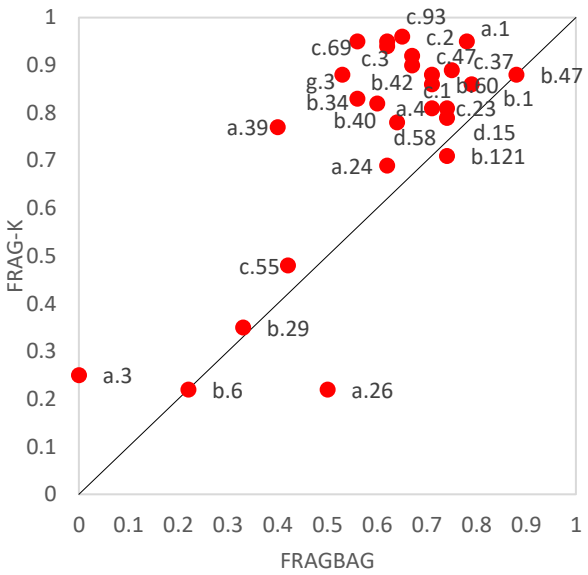
(c). 6-residue fragments



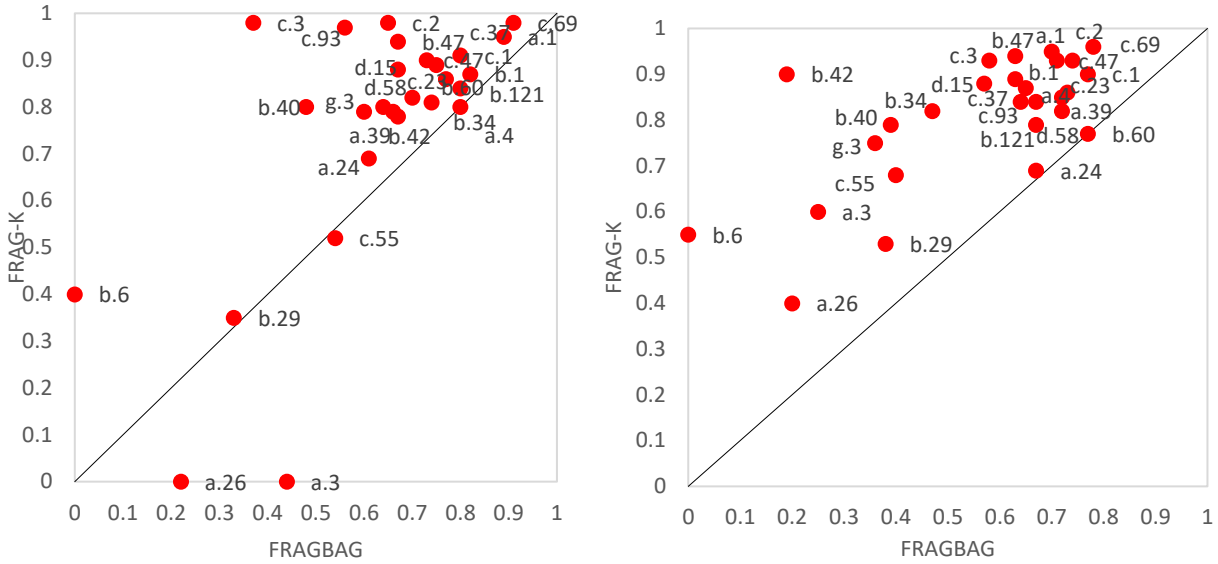
(d). 7-residue fragments



(e). 9-residue fragments



(f). 10-residue fragments



(g). 11-residue fragments (h). 12-residue fragments

Fig. 19. Comparison of classification accuracies of different fold classes using Frag-K and Fragbag fragments of different lengths as structural features in EDD dataset. The red dots represent the classification accuracies of different fold classes.

3.3.2 Structural Dictionary of Fragments with Variable Lengths

Here, we use all of the top-100 fragments in the fragment libraries of different lengths to train a random forest to classify the protein structures in EDD datasets into SCOP fold classes. A super structural feature vector is constructed to represent a protein structure, which is a concatenation of feature vectors representing fragment libraries of different lengths. Table 5 compares the 10-fold cross-validation results of precisions, recalls, and F-measures in 27 protein structure folds based on the 4- to 12-residue fragments in Frag-K as well as Fragbag. We adopt the same parameters in the random forest training procedures for both fragment libraries. Similar to the results described in Section 3.3.1, one can find that the random forest classifier trained using

Frag-K fragments as structural features yields higher overall precision, recall, and F-measure than the one using Fragbag. Indeed, the F-measure, a metric combining precision and recall, of the classifier using Frag-K is higher than that using Fragbag in almost every single SCOP fold class except for b.47, which indicates that the classifier using Frag-K demonstrates a good balance between precision and recall. Table 5 also shows the performance of random forest classifier includes longer Frag-K fragments up to 20 residues, resulting in 0.93 precision, 0.89 recall, and 0.90 F-measure in classifying all fold classes, which are higher than the classifier using only 4- to 12-residue fragments (0.85 precision, 0.79 recall, and 0.81 F-measure). This indicates that the longer fragments, which often represent the super secondary structure motifs, contribute significantly to fold classification. They are important structural keywords in the protein structure universe.

Table 5

Comparison of precision, recall, and F-measure of random forest classifiers using Frag-K and Fragbag as structure features on proteins in EDD dataset.

	Fragbag			**Frag-K**					
	L4 to L12			L4 to L12			L4 to L20		
SCOP Fold Classes	**Precision**	**Recall**	**F**	**Precision**	**Recall**	**F**	**Precision**	**Recall**	**F**
a.39	1.00	0.60	0.75	1.00	0.67	0.80	1.00	0.87	0.93
c.23	0.90	0.69	0.78	0.78	0.90	0.84	0.85	0.96	0.9
c.47	0.79	0.65	0.71	0.95	0.91	0.93	0.95	0.89	0.92
c.69	0.89	0.74	0.81	1.00	0.96	0.98	1.00	1.00	1.00
a.1	0.80	0.80	0.80	0.90	0.9	0.90	1.00	0.90	0.95
a.3	1.00	0.14	0.25	0.71	0.71	0.71	0.88	1.00	0.93
c.2	0.75	0.78	0.76	0.98	0.95	0.96	1.00	0.98	0.99
c.3	1.00	0.32	0.48	1.00	0.82	0.90	1.00	0.95	0.98
c.1	0.66	0.94	0.78	0.85	0.96	0.90	0.98	0.96	0.97

b.1	0.64	0.80	0.71	0.90	0.94	0.92	0.93	0.98	0.95
b.6	0.25	0.12	0.17	0.38	0.38	0.38	1.00	0.63	0.77
b.40	0.84	0.36	0.50	0.78	0.80	0.79	0.93	0.84	0.88
b.42	0.40	0.18	0.25	1.00	0.82	0.90	1.00	1.00	1.00
c.93	1.00	0.56	0.71	0.92	0.67	0.77	1.00	1.00	1.00
b.47	1.00	0.89	0.94	0.80	0.89	0.84	1.00	0.89	0.94
b.60	1.00	0.62	0.77	1.00	0.75	0.86	1.00	0.75	0.86
c.37	0.86	0.52	0.65	1.00	0.80	0.89	0.95	0.9	0.92
b.29	0.83	0.36	0.50	1.00	0.71	0.83	1.00	0.93	0.96
g.3	0.39	0.64	0.48	0.96	0.92	0.94	0.91	0.84	0.87
a.26	0.33	0.12	0.18	0.63	0.63	0.63	0.75	0.75	0.75
a.24	0.73	0.53	0.62	0.89	0.53	0.67	0.83	0.67	0.74
c.55	0.67	0.19	0.30	0.39	0.67	0.49	0.61	0.90	0.73
d.15	0.56	0.7	0.62	0.88	0.81	0.85	0.93	0.93	0.93
a.4	0.56	0.93	0.70	0.82	0.97	0.89	0.89	0.99	0.93
d.58	0.61	0.79	0.69	0.76	0.76	0.76	0.84	0.84	0.84
b.121	0.75	0.55	0.63	0.80	0.73	0.76	1.00	0.91	0.95
b.34	0.62	0.49	0.55	0.90	0.70	0.79	0.86	0.81	0.83
Avg/total	0.73	0.56	0.60	0.85	0.79	0.81	0.93	0.89	0.90

We rank the effectiveness of the Frag-K fragments according to the impurity decrease in the random forest classifier. Figure 20 shows the average classification precision when the top-100, … , 1600 fragments are used for the random forest classifiers. One can find that when the top-400 fragments are employed, average precision of 0.92 is achieved, although using more fragments may lead to slightly higher average precision. This means that using only 400 Frag-K fragments as structural keywords can effectively classify major SCOP folds. Figure 21 depicts the top-200 most effective Fold-K fragments for fold classification. One can find that secondary structures as well as many super-secondary structure motifs such as β-hairpins, short β-sheets, helix-loop-helix, and helix-turn-helix, are included. Figure 22 shows the distribution of the lengths of the top-200 most effective fragments, which indicates that fragments of all lengths contribute.

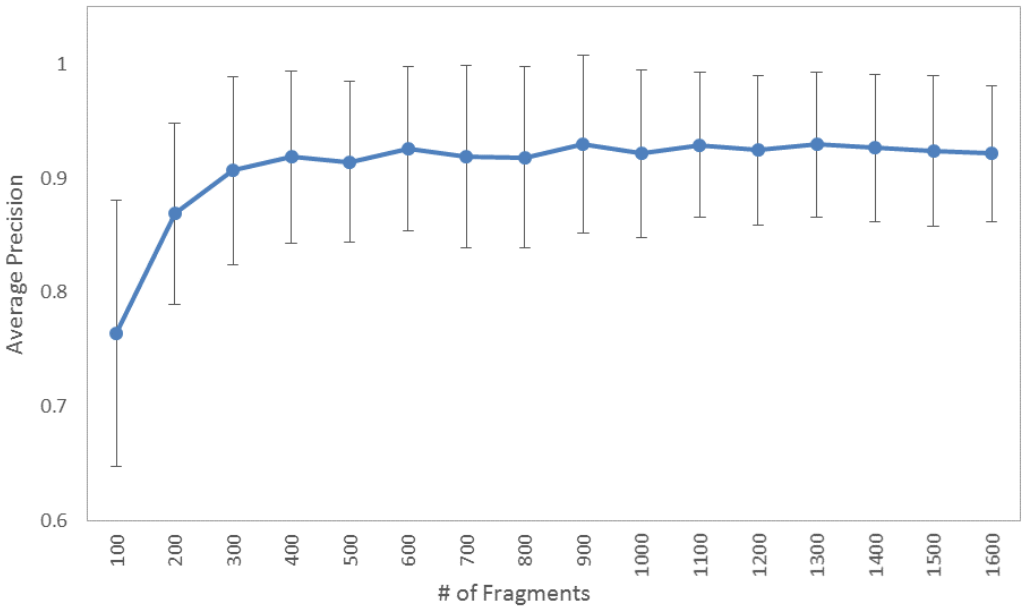


Fig. 20. Average classification precisions using top-k (ranging from 100 to 1,600) fragments.

Fig. 21. Top-200 most effective Frag-K fragments for fold classification.

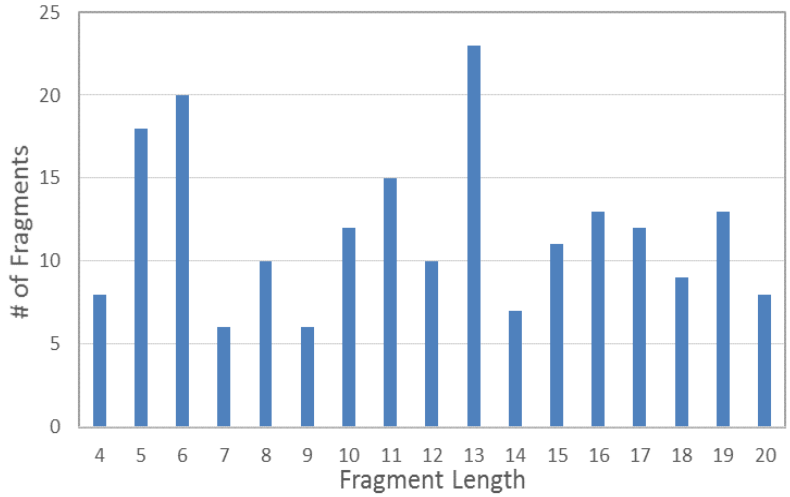


Fig. 22. Length distribution of the top-200 most effective fragment

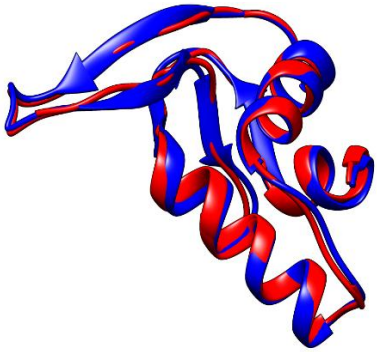
3.3.3 Assembling Protein Structure using Fragment Libraries

In addition to serving as structural keywords to distinguish folds in the structural universe, the Frag-K fragment libraries can be used to effectively assemble protein structures. The protein structure assembling process aims at generating protein backbone trace by using Frag-K fragments that can approximate the protein backbone structure with good precision. The assembly is based on the geometry of the target protein, where the amino acid label information is ignored and only its secondary structure information is used. We adopt a global fit strategy to obtain a good approximation. An iterative fragment selection procedure is performed over all possible Frag-K fragments of different lengths, where the fragments yielding a sufficiently small RMSD value compared to the original structure are favored. Starting from one end of the protein, the protein assembling process selects the most appropriate fragment that best approximates the first segment

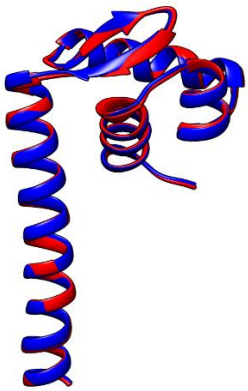
of the protein backbone. Afterward, we search the Frag-K library to build a set of feasible candidate fragments with a good local match with the already constructed segment. Typically, a good local match requires the RMSD values between the last three residues of the constructed segment and the overlapping first three residues of the selected fragments are within a certain threshold. Then, we select the fragment from the feasible candidate set yielding the minimum RMSD value with respect to the corresponding segment in the target structure to extend the constructed segment. If no feasible fragments are found, the one with minimum RMSD to the corresponding segment in the target structure is selected. The fragment assembling process is repeated until the complete protein backbone trace is generated.

Figure 23 displays the backbone traces of several protein structures by Frag-K fragments with variable lengths. These protein structures belong to different fold classes. One can find that all assembled structures yield resolutions less than 2A. This indicates that the Frag-K fragments can be used effectively as a reduced representation of native protein structures, which can be applied to a wide variety of applications such as ab initio protein structure modeling [67], protein loop modeling [68], and protein design [69].

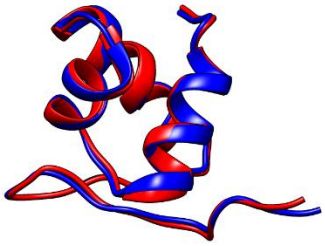
d4maka_ d.58: Ferredoxin-like
alpha+beta sandwich with antiparallel beta-sheet; (beta-alpha-beta), 0.51Å.



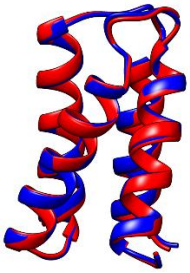
d4j20a_ a.3: Cytochrome
core: 3 helices; folded leaf, opened, 0.64Å.



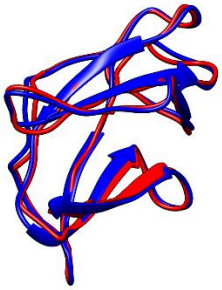
d1dp7p_ a.4: DNA/RNA-binding 3-helical bundle
core: 3-helices; bundle, closed or partly opened, right-handed twist; up-and down, 1.41Å.



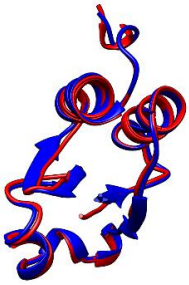
d1r7ja_ a.4: DNA/RNA-binding 3-helical bundle
core: 3-helices; bundle, closed or partly opened, right-handed twist; up-and down, 1.27Å



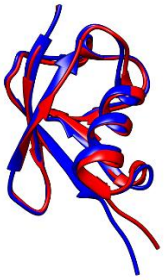
d2ve8a_ a.4: DNA/RNA-binding 3-helical bundle
core: 3-helices; bundle, closed or partly opened, right-handed twist; up-and down, 0.40Å.



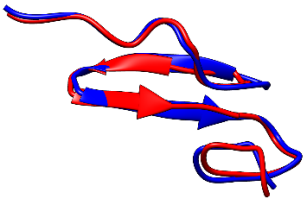
d1ls1a1 a.24: Four-helical up-and-down bundle
core: 4 helices; bundle, closed or partly opened, left-handed twist; up-and-down, 0.39Å.



d3uzqb_ b.1: Immunoglobulin-like beta-sandwich
sandwich; 7 strands in 2 sheets; Greek-key, 1.78Å



d3eina1 c.47: Thioredoxin fold *core: 3 layers, a/b/a; mixed beta-sheet of 4 strands, order 4312; strand 3 is antiparallel to the rest, 0.65Å.*



d3phxb d.15: beta-Grasp (ubiquitin-like) *core: beta(2)-alpha-beta(2); mixed beta-sheet, 0.78Å.*

d1edmb_ g.3:knottins (small inhibtors, toxins, lectins), *disulfide-bound fold; contains beta-hairpin with two adjacent disulfides, 0.33Å.*

Fig. 23. Approximations of 10 protein structures using 4- to 20-residue Frag-K fragments. The native is in blue and the assembled structure is in red.

3.4 Summary

In this chapter, we apply the randomized spectral clustering algorithm to process large-scale protein backbone fragment sets derived from the continuously growing PDB to generate Frag-K libraries containing 4- to 12-residue protein fragments. The Frag-K libraries are used as structural features to encode protein structures. We train random forests based on Frag-K fragments to classify major SCOP folds. Our results show that using about 400 4- to 12-residue fragments as structural keywords, one can classify major SCOP folds with high accuracy.

The Frag-K fragment libraries are deposited at http://hpcr.cs.odu.edu/FragK/. Frag-K can also be used to investigate interactions between fragments [70], study motif formations in protein families, monitor structural keywords formation during protein folding process, and de novo protein structure design.

CHAPTER IV

DEEPFRAG-K: A FRAGMENT-BASED DEEP LEARNING APPROACH FOR PROTEIN FOLD RECOGNITION

In this chapter, we present a novel deep neural network architecture, so-called DeepFrag-k, to classify target protein sequences into known protein folds. The fundamental idea is to convert a target protein sequence into structural fragments that popularly exist in protein structures [71], represented as a fragment vector, which contains highly discriminative features to distinguish the protein fold [72]. Deep-Frag-k is composed of two stages. The first stage uses a multi-modal Deep Belief Network (DBN) to fuse multiple groups of features, including sequence composition, amino acid physicochemical properties, and evolutionary information, to precisely predict potential structure fragments for a given sequence, which are represented as a fragment vector. Then, a 1-D Convolution Neural Network (CNN) is employed to classify the fragment vector into the appropriate fold.

4.1 Methodology

4.1.1 DeepFrag-k Fold Recognition Architecture

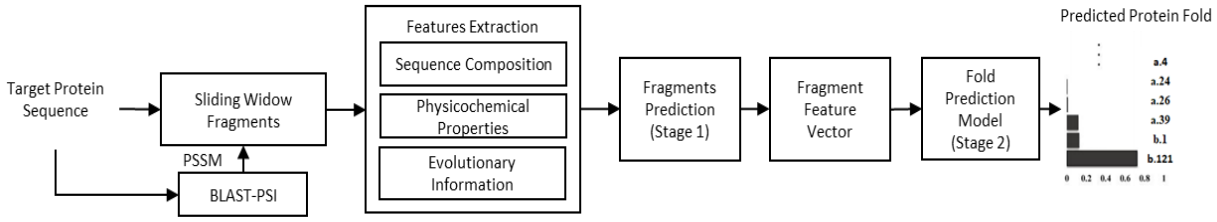


Fig. 24. Two-stage protein fold recognition architecture.

Figure 24 presents the two-stage deep neural network architecture of DeepFrag-k. In the first stage, we predict a fragment vector representation of a target protein sequence using a fragment prediction model based on multimodal DBN [73], which predicts the potential fragments that the target protein sequence will form during protein folding process. In particular, we focus on the top-100 most popular fragments, with 4- to 20-residue in length, described in our Frag-K fragment libraries [71, 72]. Our results in section 3.3 show that these fragments can be used as the structural "keywords" to effectively distinguish between major protein folds. In the multimodal DBN, the DBNs interact with each other to learn fragment latent representation on the set of features derived from sequence composition, physicochemical properties, and evolutionary information. The output of the first stage is a fragment vector with respect to the target protein sequence. Afterwards, in the second stage, this fragment vector is fed to a 1D Convolution Neural Network (1D-CNN) [74, 23] classifier, as the feature vector of the target protein sequence, to predict the likeliness of the protein folds.

4.1.1.1 Fragment Prediction *(Stage 1)*

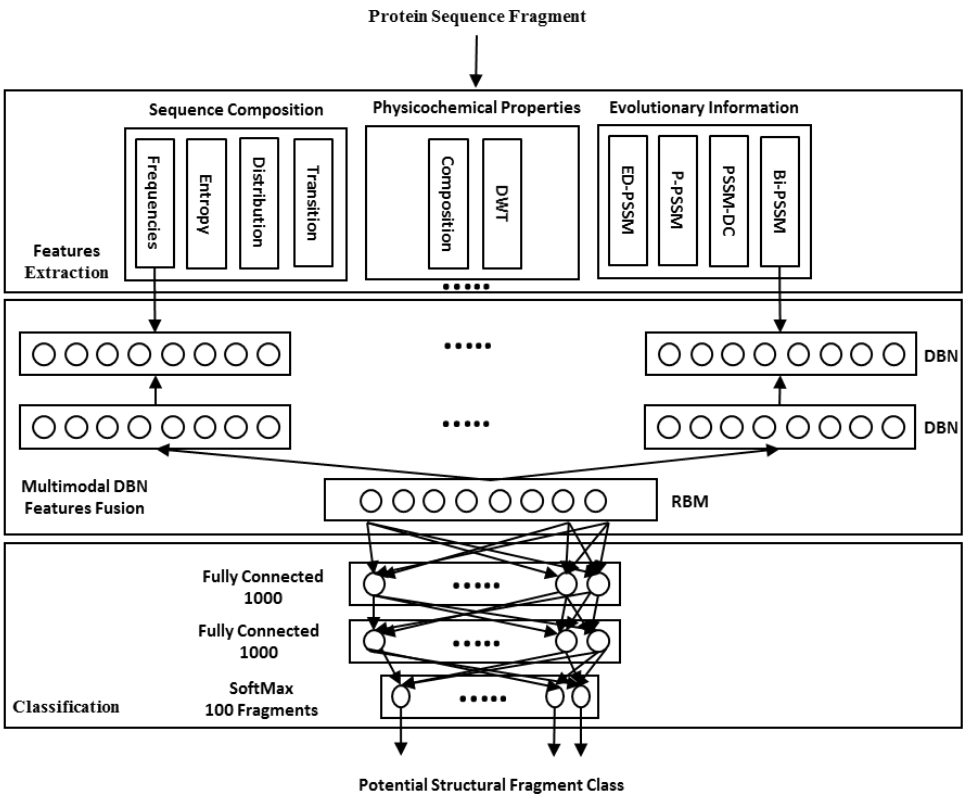


Fig. 25. Fragments prediction multimodal DBN architecture.

A protein fold distinguish itself by forming certain unique secondary structures and super-secondary structure motifs, such as β-hairpins, short β-sheets, helix-loop-helix, and helix-turn-helix, which are represented as structural fragments. Correctly predicting these fragments from a given sequence can lead to effective features for fold recognition. However, the sequence features to predict fragments hold distinct statistical properties and the correlations between them are highly non-linear [75]. For a shallow model, it is difficult to capture these correlations and form an integrated informative representation. Our fragment prediction model consists of a multimodal DBN and a fully-connected network. Our motivation for the pro-posed multimodal DBN is to

tackle the above challenge by using an integrated representation to enhance the fragment prediction accuracy [73, 23]. Figure 25 summarizes the framework of our proposed fragment prediction model. We use the Frag-K fragment libraries to train the fragment prediction model. First, we use the extracted sequence composition [76], physicochemical properties [76], and evolutionary information [76, 77, 78, 79] as feature groups to learn the latent representations of the top-100 Frag-K fragments. As shown in section 3.3, the top-100 Frag-K fragments are capable of classifying major SCOP folds in high accuracy and can also be used to assemble most protein structures in high precision. The multiple feature representations learned by the DBNs are concatenated to train a Restricted Boltzmann Machine (RBM) model [73] to fuse a latent feature representation for the feature groups. Finally, two fully-connected 1,000x1,000 neural network layers followed by a SoftMax layer of 100 output nodes, representing the top-100 Frag-K fragments, are trained with these latent feature representations to make the fragment prediction. Such layer-by-layer learning helps gradually extract the effective features from the original feature groups [80]. The multimodal DBN learns discriminative latent features as a joint distribution determined by the hidden variables of non-correlated feature groups input [73]. As a result, the hybrid framework of multi-modal learning fuses an abstraction level representation, which enables the fragment predictor to integrate different feature groups for fragments of different lengths flexibly.

The training of the fragment prediction model is performed via Stochastic Gradient Descent method. During the training process, the Frag-K fragment library, with 1,000 samples in each fragment class, is randomly split into batches, each of which contains 500 samples. In order to prevent overfitting, dropout layers are inserted after every hidden layer with 0.5 dropout rate and an early stop-ping strategy is employed.

4.1.1.2 Fold prediction (Stage 2)

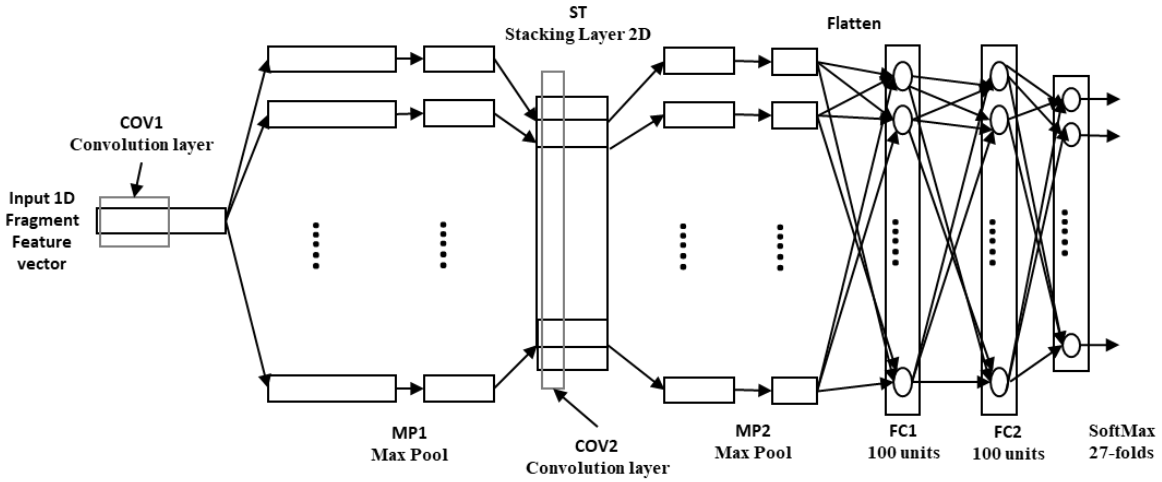


Fig. 26. Protein Fold Classification 1D-CNN model.

The fragment feature vector generated from stage 1 is fed to a 1D-CNN architecture to predict protein fold, as shown in Figure 26. The proposed 1D-CNN comprises two pairs of convolutional and max pooling layers (COV1-MP1 and COV2-MP2), two fully-connected layers FC1 and FC2, and a SoftMax layer. Between MP_1 and COV_2, we include a stacking layer ST. The COV1 layer contains 10 convolution filters, producing 10 filtered versions of the fragment feature

vector as output. These filtered versions are then subsampled in max pooling layer MP1. The stacking layer rearranges the output of MP1 so that a 2D stack of the generated features from COV1 is sent to the second convolutional layer COV2. The convolution filters in COV2 are 2D filters, with the same height as the ST layer. The purpose of these 2D filters is to capture the relationships across the latent features produced by the convolution filters of the original fragment vector in COV1. Then the generated output is subsampled in max pooling layer MP2. In order to classify the flattened output of MP2 into corresponding folds, two fully-connected layers, FC1 and FC2, followed by a SoftMax layer are employed.

4.1.2 Feature Extraction

Table 6

Protein sequence features.

Feature	Type	Dimension
Sequence Composition	Frequency of Function Group	10
	Information Entropy	2
	Distribution	20
	Transition	45
Physicochemical properties	Pseudo Amino Acid Composition	40
	Discrete Wavelet Transformation	42
Evolutionary Information	P-PSSM	400
	PSSM-DC	400
	Bi-Gram PSSM	400
	ED-PSSM	400

Constructing a proper feature vector from proteins is a key step for a successful protein fragment prediction [77, 81]. Using multiple features extraction strategy, representing sequence, evolutionary, physicochemical information of a protein sequence fragment, maximizes the

discriminative capability of the fold recognizer [82, 83]. The sequence features for fragments used in DeepFrag-k include frequencies of functional groups, information entropy of amino acids and dipeptides [84], distribution of amino acids relative positions [83], and transitions of functional groups [85]. The physicochemical features include PseAAC (Pseudo Amino Acid Composition) [86, 87] and Discrete Wavelet Transform (DWT) [88] of hydrophobicity, flexibility, and average accessible surface area of amino acids in a fragment. The evolutionary features are described by various forms of position-specific scoring matrix (PSSM) profiles [35] including profile PSSM (P-PSSM), PSSM-Dipeptide Composition (PSSM-DC) [76], Bi-gram PSSM (Bi-PSSM) [34], and Evolutionary Difference-PSSM (ED-PSSM) [89]. These features are summarized in Table 6.

4.2 Results

4.2.1 Fragment Prediction Model

The extracted sequence composition, physicochemical properties, and evolutionary information features of the Frag-K fragments are fed to the fragment prediction model to predict their potential corresponding fragments classes. We investigate the performance of the classifier measured by specificity, sensitivity, and accuracy, which are defined as the percentage of predicted fragment classes that are true positives, the percentage of true positives that are correctly predicted, and the fraction of fragments that are correctly classified, respectively.

We first examine the classification of sequence fragments of the same length. Figure 27 shows the accuracy, specificity, and sensitivity of the ten-fold cross-validation results for top-100 Frag-K fragment targets of each length, ranging from 4 to 20 residues. One can find that the prediction accuracies of longer fragments (≥10 residues) are better than those of the shorter ones, where both specificity and sensitivity are over 80%. This is due to the fact that the longer fragments encompass richer discriminative information. However, when the top-100 Frag-K fragments with variable lengths are used as the target classes, the prediction accuracy reaches over 90%, because these top-100 Frag-K fragments with variable lengths are more representative structural keywords in the protein structure universe, as we showed in section 3.3.

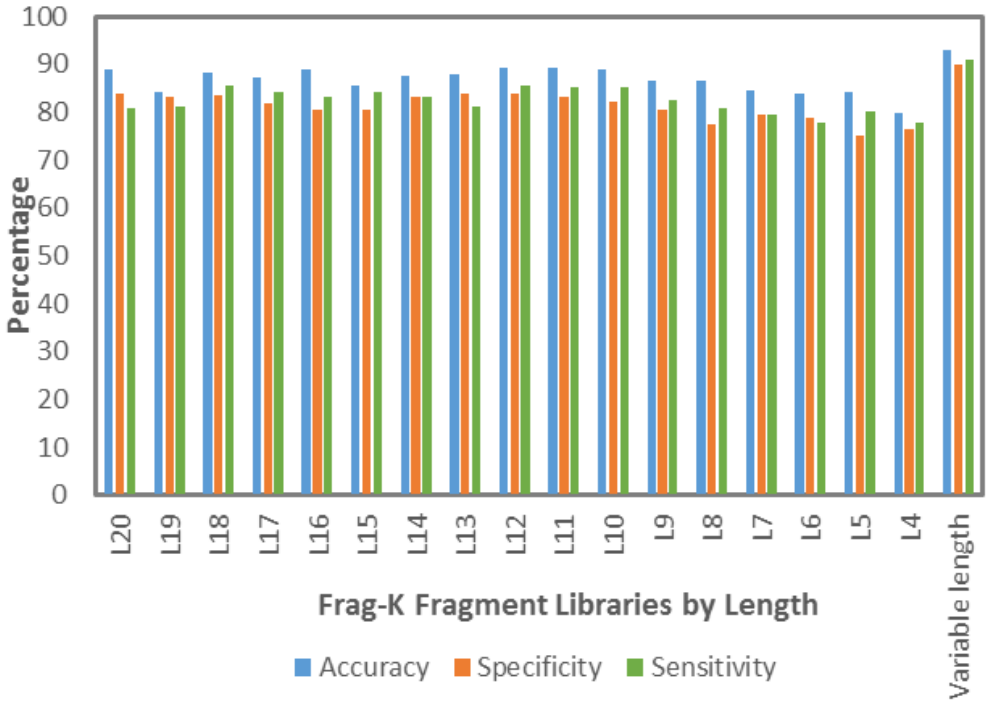


Fig. 27. Accuracy, specificity, and sensitivity of fragment libraries models.

We analyze the effectiveness of the three feature groups (Table 6) used to represent the sequence fragments on variable length Frag-K fragment prediction accuracy. We compose individual and combined sequence composition, physicochemical properties, and evolutionary information feature vectors to train the fragment prediction model showed in Figure 55. The ten-fold cross-validation accuracy results are reported in Figure 28. The evolutionary information plays the most important role; however, all of these feature groups contribute to fragment accuracy improvements.

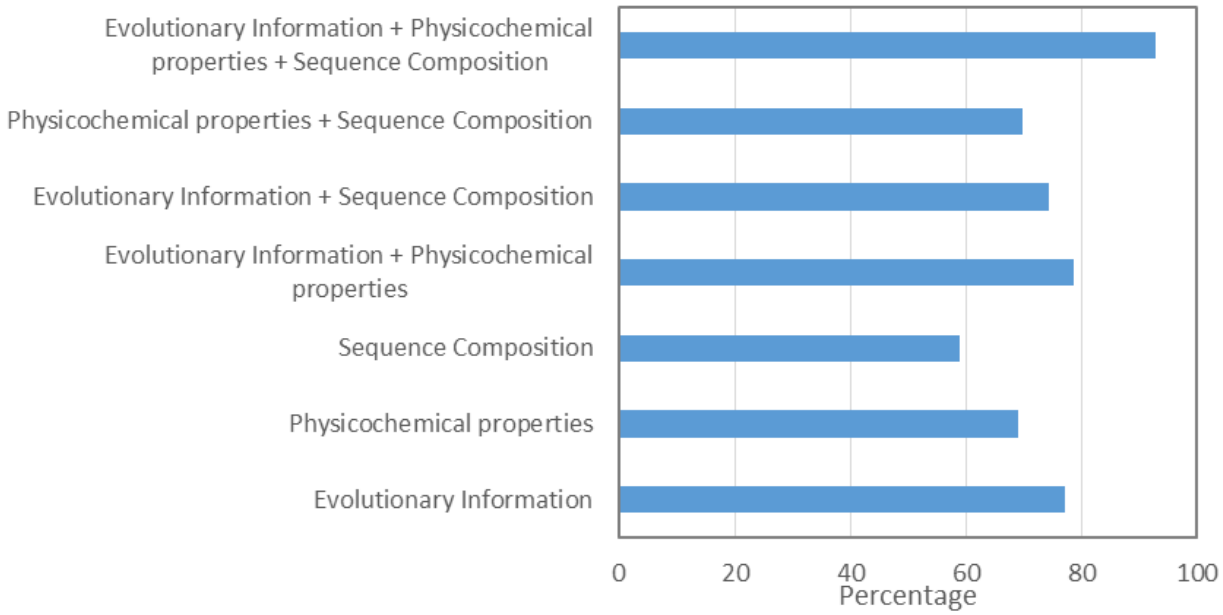


Fig. 28. Accuracy of variable length Frag-K fragment prediction when different feature groups and their combinations are applied.

In section 3.3.2 it is indicated that the Frag-K variable length fragment library achieves higher fold classification accuracy than fixed length fragment library over EDD dataset. This demonstrates that the diversity of the fragments representing the super secondary structure motifs contributes significantly to fold classification. Additionally, it is established in section 3.3.3 that the Frag-K variable length fragment library can be used effectively to assemble the protein backbone trace with good precision. The Frag-K variable length fragment library can be used with a global fit strategy to obtain a good approximation of a target protein. The higher classification accuracy and the ability to reconstruct protein backbone trace of Frag-K variable length fragment library are due to its selection and ranking methodologies which are explained in section 3.1.4.

4.2.2 Fold Classification Model

As shown in section 3.3, the Frag-K fragment library with variable length achieves higher fold classification accuracy than fixed-length ones. Moreover, our results in the previous section show that the prediction accuracy on variable length Frag-K fragments is higher than individual fixed-length fragments. Therefore, we used the fragment vectors based on variable-length fragment predictions from the fragment prediction model for the fold recognition model.

We use the sequences in DD, EDD, and TG datasets to evaluate the performance of DeepFrag-k. First, for a given sequence, we use a sliding window of 4 to 20 residues to consecutively segment it into a set of overlapping fragments, where gaps and non-protein residues are excluded. Figure 29 and Figure 30 compare the fold recognition accuracy of DeepFrag-k with

other fold recognition methods, including PFP-Pred [46], GAOEC [49], ThePFP-FunDSeqE [39], Dehzangi et al. [90, 91], MarFold [51], PFP-RFSM [37], Feng and Hu [53], Feng et al. [54], PFPA [40], Paliwal et al. [92, 93], Dehzangi et al. [94], HMMFold [95], Saini et al. [87], Lyons et al. [96], and Profold [42] in protein fold recognition.

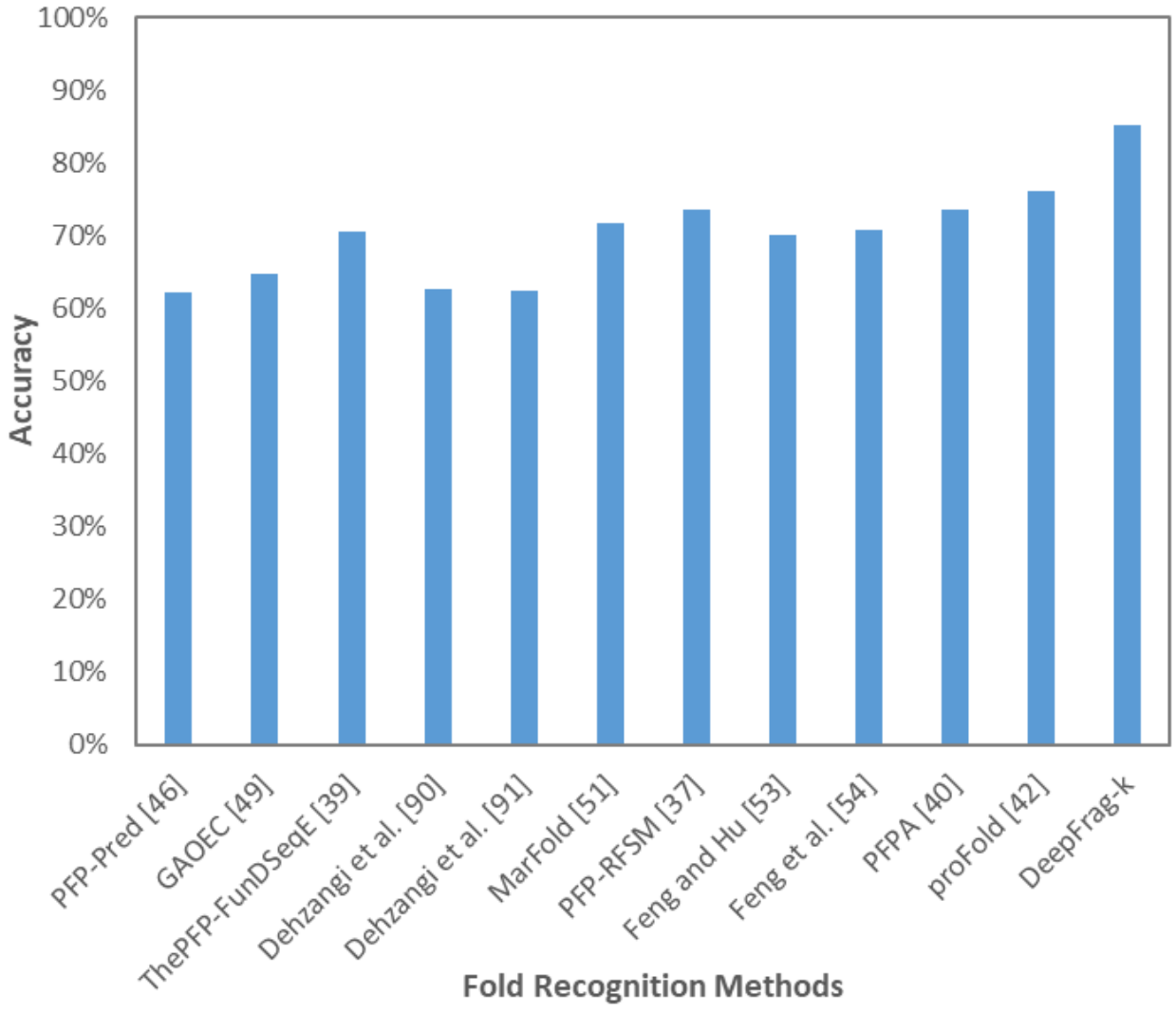


Fig. 29. Comparison with existing ensemble learning methods on DD-dataset.

Figure 29 summarizes the ten-fold cross-validation results of DeepFrag-k and other fold recognition methods on the DD dataset. DeepFrag-k outperforms the other methods by yielding 85.3% accuracy, which is 9.1% higher than the second highest, proFold (76.2%). More detailed comparisons between DeepFrag-K and ProFold for each individual protein fold are listed in Table 7. One can find that DeepFrag-k demonstrates better fold recognition accuracy than ProFold in 18 out of 27 protein folds. It is also important to notice that DeepFrag-k shows more balanced prediction accuracy. In particular, for the folds, such as b.34, b.47, c.3, c.37, and d.15, that ProFold exhibits poor prediction results, DeepFrag-k yields significant improvements.

Table 7

DeepFrag-K and ProFold folds classifications accuracies for DD-dataset.

#	Fold ID	Fold Name	Accuracy	
			DeepFrag-K	ProFold
1	a.1	Globin-like	98	100
2	a.3	Cytochrome c	95	100
3	a.4	DNA/RNA-binding 3-helical bundle	85.9	60
4	a.24	4-Helical up-and-down bundle	91.5	87.5
5	a.26	4-Helical cytokines	98.9	88.9
6	a.39	EF hand-like	90.8	77.8
7	b.1	Immunoglobulin-like β-sandwich	91.1	84.1
8	b.6	Cupredoxin-like	78.7	66.7
9	b.121	Nucleoplasmin-like/VP	91.3	92.3

10	b.29	ConA-like lectins/glucanases	76.7	66.7
11	b.34	SH3-like barrel	78	50
12	b.40	OB-Fold	80.4	68.4
13	b.42	β-Trefoil	89	100
14	b.47	Trypsin-like serine proteases	75	50
15	b.60	Lipocalins	90.5	100
16	c.1	TIM β/α-barrel	93.8	93.8
17	c.2	FAD/NAD(P)-binding domain	89.7	91.7
18	c.3	Flavodoxin-like	60.2	46.2
19	c.23	NAD(P)-binding Rossmann	90.2	85.2
20	c.37	P-loop containing NTH	79.5	50
21	c.47	Thioredoxin-fold	97.5	87.5
22	c.55	Ribonuclease H-like motif	75.3	58.3
23	c.69	α/β-Hydrolases	78.4	71.4
24	c.93	Periplasmic binding protein-like	92	100
25	d.15	β-Grasp (ubiquitin-like)	69.35	25
26	d.58	Ferredoxin-like	76.8	59.3
27	g.3	Knottins (small inhibitors, toxins, lectins)	88.2	96.3
Accuracy			85.25	76.18

We further evaluate the performance of DeepFrag-k on the EDD and TG datasets. The ten-fold cross-validation results in comparison with other methods are illustrated in Figure 30. DeepFrag-k yields 96.1% and 97.5% accuracies on EDD and TG datasets, respectively, which are

higher than the other fold recognition methods. Due to significantly more samples are available in EDD and TG datasets, which is particularly helpful for our deep learning model to capture the discriminative features of the protein folds in sequence space, the DeepFrag-k yields better fold recognition accuracies in EDD and TG datasets than that in DD dataset.

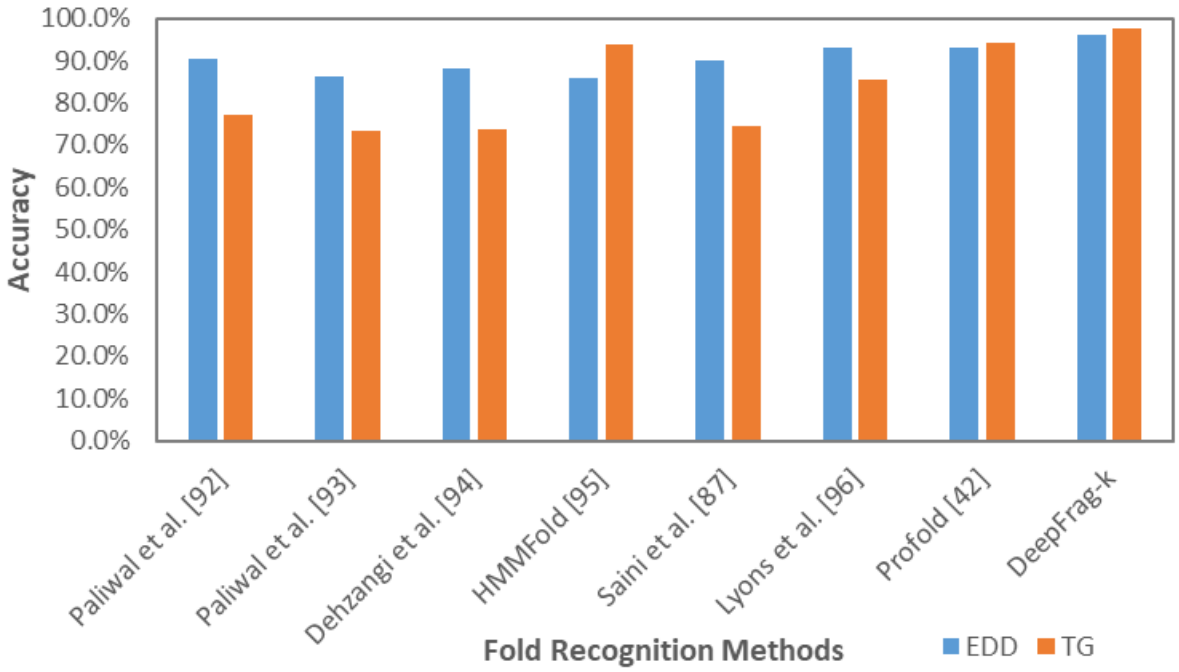


Fig. 30. Comparing DeepFrag-k with other fold recognition methods on the TG and EDD datasets.

Figure 31 depicts the Class Activation Map (CAM) [97, 98] of DeepFrag-k on the EDD dataset to show how protein folds classified based on the fragment feature vectors from the protein sequences. The activation units that are most discriminative to fold classifications are identified, which are highly weighted. The combination of these class-specific units guides DeepFrag-k in distinguishing each fold. One can observe that the fold classification model makes use of more

activation units to classify α/β or α+β proteins (C.1 to C.93), when compared to all α (A.1 to A.39) and all β proteins (B.1 to B.60). However, in folds of small proteins, such as G.3, only a few activation units are effective in the fold recognition process.

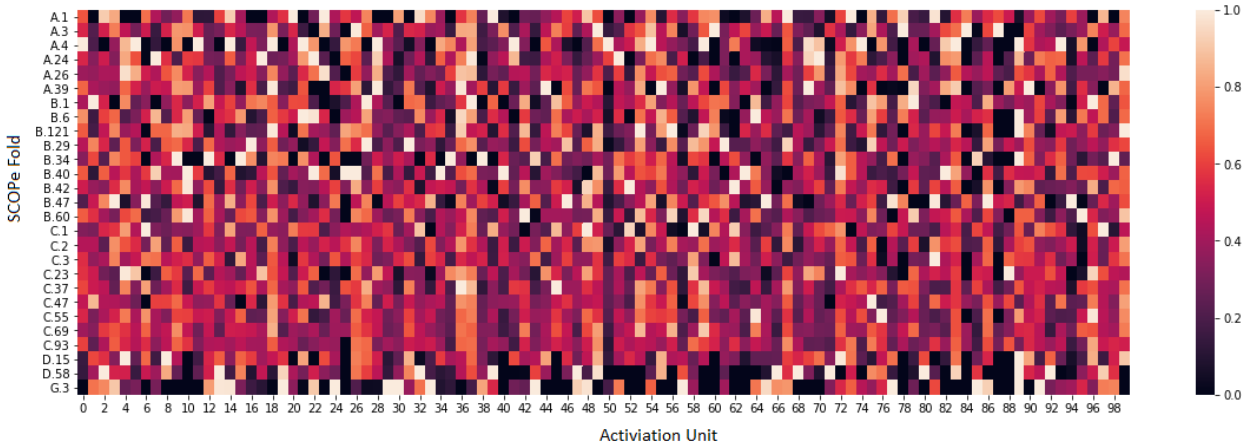


Fig. 31. EDD fold classification class activation map.

4.3 Summary

In this chapter, we design DeepFrag-k, a two-stage deep learning neural network architecture, for fold recognition. The fragment prediction stage derives effective fragment feature vectors by fusing sequence composition, physicochemical properties, and evolutionary information features groups of sequence fragments to the fold recognition stage. Due to the discriminative capability of the fragment feature vectors, Deep-Frag-k yields significant accuracy enhancement compared to other fold recognition methods on the DD, EDD, and TG datasets.

The features derived in DeepFrag-k are based on sequence fragments. They can be incorporated with other sequence or structure features [99], such as inter-residue interactions [81],

to further improve fold recognition. This will be our future research direction. The DeepFrag-k package can be downloaded at http://hpcr.cs.odu.edu/deepfragk.

CHAPTER V

CONCLUSION AND FUTURE WORK

5.1 Conclusion

Protein folding is one of the major research areas in the bioinformatics field. Despite, the progress in protein fold research, there is a huge need for more work. Hence, the processes of fold formation and stabilization are still not fully understood. One of the important factors to correctly recognize the protein fold is the prediction of local backbone conformations. The favorable local backbone conformations can be carefully extracted to predict the conformation of a new sequence. Various methods are proposed for an efficient prediction of local backbone conformations. Accordingly, it is becoming increasingly clear that these methods can contribute significantly to improve the accuracy of recognizing related folds.

In this work, we apply the randomized spectral clustering algorithm to process large-scale protein backbone fragment sets derived from the continuously growing PDB to generate Frag-K libraries containing 4- to 20-residue protein fragments. The Frag-K libraries are used as structural features to encode protein structures. We train random forests based on Frag-K fragments to classify major SCOP folds. Our results show that using about 400 4- to 20-residue fragments as structural keywords, can classify major SCOP folds with high accuracy.

Additionally, we design DeepFrag-k, a two-stage deep learning neural network architecture, for fold recognition. The fragment prediction stage derives effective fragment feature vectors by fusing sequence composition, physicochemical properties, and evolutionary information features groups of sequence fragments to the fold recognition stage. Due to the discriminative capability of the fragment feature vectors, Deep-Frag-k yields significant accuracy enhancement compared to other fold recognition methods on the DD, EDD, and TG datasets.

5.2 Future Work

One of the most important reactions in biology is protein folding. Since the discovery that proteins can fold naturally without outside help, an intensive work of research in protein folding has been done. However, the primary questions about protein folding are still not answered, such as: How do proteins fold? Why do they fold in that way? These questions are significantly important for protein science and its various applications. A large literature has been generated over the years based on these questions leading to different models for the folding process. Additionally, the advances in computational methods add a new perspective.

We plan to consider Frag-K libraries and DeepFrag-k to answer the central questions of protein folding (how, why, and the encoding problem). There are several interesting aspects that we would like to explore. For instance, it would be interesting to monitor structural fragments formation during protein folding process in order to study the fold formation by analyzing the

motif dynamics in protein folding simulation. These will be our future research directions which shall provide important insights in the protein folding pathways.

We expect that the proposed protein fragments libraries and protein folding recognition framework will lead to the discovery of more accurate and informative protein folding pathways. Also, they will be used to improve the understanding of various important steps of protein folding process ranging from template identification, alignment, and quality assessment by taking advantage of the continuous growth of protein sequence and structural database in the era of "Big Data". Furthermore, we propose to rely on the solid ground of experiment rather than the often used suggestions that are less-definitive.

PUBLICATIONS

1. Wessam Elhefnawy, and Yaohang Li. “DeepFrag-k: A Fragment-based Deep Learning Approach for Protein Fold Recognition”,in 15th International Symposium on Bioinformatics Research and Applications, ISBRA 2019, 2019. (submitted)
2. Wessam Elhefnawy, Yaohang Li,” Decoding the Structural Keywords in Protein Structure Universe ”, Journal of Computer Science and technology, 2019.
3. Maha Abdelrasoul, Komala Ponniah, Alice Mao, Meghan Warden, Wessam Elhefnawy, Yaohang Li, Steven Pascal, “Conformational Clusters of Phosphorylated Tyrosine”, Journal of the American Chemical Society, 2017.
4. Elhefnawy, Wessam, Lin Chen, Yun Han, and Yaohang Li. "ICOSA: A distance-dependent, orientation-specific coarse-grained contact potential for protein structure modeling." Journal of molecular biology 427, no. 15 (2015): 2562-2576.
5. Elhefnawy, Wessam, Min Li, Jianxin Wang, and Yaohang Li. "Construction of Protein Backbone Fragments Libraries on Large Protein Sets Using a Randomized Spectral Clustering Algorithm.", In International Symposium on Bioinformatics Research and Applications, pp. 108-119. Springer, Cham, 2017.

6. Wessam Elhefnawy, Adam Boudion, Erich O'Saben, Maha Abdelaal, Steven M. Pascal, and Yaohang Li1. "Structural Analysis and Prediction of Protein Phosphorylation Sites",in 11th International Symposium on Bioinformatics Research and Applications, ISBRA 2015, June 7-10, 2015.
7. Wessam Elhefnawy, " Make My Neighborhood Safe Again: An Agent Based Model for Burglary Crime Prediction and Capture It's Patterns " Proceedings of Modeling, Simulation, and Visualization Student Capstone Conference, April 14th, 2017. (Best Paper Award)
8. Wessam Elhefnawy, Jesse Wright, Kumara Kallepalli, Kevin Racheal, Ajay Gupta, Yaohang Li, Rohit Parimi, Parth Shah," What Differentiates News Articles with Short and Long Shelf Lives? ", 6th IEEE International Conference on Big Data and Cloud Computing (BDcloud 2016), October 8th -10th , 2016.
9. Wessam Elhefnawy, Li Chen, Yaohang Li. "Weighted Path Spectral Clustering For Image Segmentation". Proceedings of Modeling, Simulation, and Visualization Student Capstone Conference, April 14th, 2016.
10. Wessam Elhefnawy, Yaohang Li and Li Chen "Improving Normalize-Cut Image Segmentation using λ-connectedness", In 4th annual conference of The Extreme Science and Engineering Discovery Environment, XSEDE15, July 26 -30, 2015

11. Wessam Elhefnawy, and Yaohang Li “Connectedness-Cut for Image Segmentation”, Proceedings of Society for industry and Applied Mathematics SIAM, April 11, 2015.

REFERENCES

[1] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov and P. E. Bourne, "The protein data bank," *Nucleic acids research,* vol. 28, no. 1, pp. 235-242, 2000.

[2] N. K. Fox, S. E. Brenner and J.-M. Chandonia, "SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures," *Nucleic acids research,* vol. 42, no. D1, pp. D304-D309, 2013.

[3] C. R. Woese and G. E. Fox, "Phylogenetic structure of the prokaryotic domain: the primary kingdoms," *Proceedings of the National Academy of Sciences 74,* vol. 11, pp. 5088-5090, 1977.

[4] D. DasGupta, "An overview of artificial immune systems and their applications Springer, Berlin, Heidelberg, .," *Artificial immune systems and their applications,* pp. 3-21, 1993.

[5] P. H. Raven, R. F. Evert and S. E. Eichho, Biology of plants, Macmillan, 2005.

[6] P. Y. Chou and G. D. Fasman, "Conformational parameters for amino acids in helical, β-sheet, and random coil regions calculated from proteins," *Biochemistry,* vol. 13, no. 2, pp. 211-222., 1974.

[7] S. S. Mader, Concepts of biology, Mcgraw-hill, 2009.

[8] S. T. Rao and M. G. Rossmann, "Comparison of super-secondary structures in proteins," *Journal of molecular biology,* vol. 76, no. 2, pp. 241-256., 1973.

[9] "Ribonuclease," Wikipedia, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Ribonuclease. [Accessed 07 December 2017].

[10] "RCSB PDB," Rcsb.org, [Online]. Available: https://www.rcsb.org/pdb/static.do?p=general_information/pdb_statistics/index.html.. [Accessed 7 December 2017].

[11] C. Levinthal, "How to fold graciously," *Mossbauer spectroscopy in biological systems,* vol. 67, pp. 22-24, 1969.

[12] C. Levinthal, "Are there pathways for protein folding," *Journal de chimie physique,* vol. 65, pp. 44-45, 1968.

[13] B. Webb and A. Sali , "Protein structure modeling with MODELLER," *Protein Structure Prediction* , pp. 1-15, 2014.

[14] C. H. Ding and I. Dubchak, "Multi-class protein fold recognition using support vector machines and neural networks," *Bioinformatics* , vol. 17, no. 4, pp. 349-358, 2001.

[15] S. R. Eddy, "Profile hidden Markov models," *Bioinformatics,* vol. 14, no. 9, pp. 755-763, 1998.

[16] J. Gough, K. Karplus, R. Hughey and C. Chothia, "Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure," *Journal of molecular biology,* vol. 313, no. 4, pp. 903-919, 2001.

[17] M. J. Sippl, "Knowledge-based potentials for proteins," *Current opinion in structural biology* , vol. 5, no. 2, pp. 229-235, 1995.

[18] Y. Li, H. Liu, I. Rata and E. Jakobsson, "Building a knowledge-based statistical potential by capturing high-order inter-residue interactions and its applications in protein secondary

structure assessment," *Journal of chemical information and modeling,* vol. 53, no. 2, pp. 500-508, 2013.

[19] P. J. Hajduk and J. Greer, "A decade of fragment-based drug design: strategic advances and lessons learned," *Nature reviews Drug discovery,* vol. 6, no. 3, pp. 211-219, 2007.

[20] D. Xu and Y. Zhang, "Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field," *Proteins: Structure, Function, and Bioinformatics,* vol. 80, no. 7, pp. 1715-1735, 2012.

[21] P. Larranaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano and e. al, "Machine learning in bioinformatics," *Briefings in bioinformatics,* pp. 86-112, 2006.

[22] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior and e. al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine,* vol. 29, no. 6, pp. 82-97, 2012.

[23] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature,* vol. 521, no. 7553, pp. 436-444, 2015.

[24] Y. LeCun, "LeNet-5, convolutional neural networks," 2015. [Online]. Available: URL: http://yann. lecun. com/exdb/lenet. [Accessed 5 December 2018].

[25] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size," *arXiv preprint arXiv1602.07360,* 2016.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[27] C. Szegedy, S. Ioffe, V. Vanhoucke and A. A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *AAAI*, pp. 4278-4284, 2017.

[28] B. Leo, J. Friedman, C. J. Stone and R. A. Olshen, Classification and regression trees, CRC press, 1984.

[29] A. Verikas, A. Gelzinis and M. Bacaus, "Mining data with random forests: A survey and results of new tests," *Pattern Recognition*, vol. 44, no. 2, pp. 330-349, 2011.

[30] Y. H. Taguchi and M. M. Gromiha, "Application of amino acid occurrence for discriminating different folding types of globular proteins," *BMC bioinformatics*, vol. 8, no. 1, p. 404, 2007.

[31] Q. Dong, S. Zhou and J. Guan, "A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation," *Bioinformatics*, vol. 20, no. 2655-2662, p. 25, 2009.

[32] M. T. A. Shamim, M. Anwaruddin and H. A. Nagarajarm, "Support vector machine-based classification of protein folds using the structural properties of amino acid residues and amino acid residue pairs," *Bioinformatics*, vol. 23, no. 24, pp. 3320-3327, 2007.

[33] J. Yang and X. Chen, "Improving taxonomy-based protein fold recognition by using global and local features," *Proteins: Structure, Function, and Bioinformatics*, vol. 79, no. 7, pp. 2053-2064, 2011.

[34] A. Sharma, J. Lyons, A. Dehzangi and K. K. Paliwal, "A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition," *Journal of theoretical biology*, vol. 320, pp. 41-46, 2013.

[35] S. F. Altschul, T. L. Madden, . A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller and D. J. Lipman. , "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic acids research,* vol. 25, no. 17, p. 3389, 1997.

[36] T. Damoulas and M. A. Girolami, "Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection," *Bioinformatics,* vol. 24, no. 10, pp. 1264-1270, 2008.

[37] J. Li, J. Wu and K. Chen, "PFP-RFSM: protein fold prediction by using random forests and sequence motifs," *Journal of Biomedical Science and Engineering,* vol. 6, no. 12, p. 1161, 2013.

[38] C. Lampros, T. Simos, T. P. Exarchos, K. P. Exarchos, C. Papaloukas and D. I. Fotiadis, "Assessment of optimized markov models in protein fold classification," *Journal of bioinformatics and computational biology,* vol. 12, no. 4, p. 1450016, 2014.

[39] H.-B. Shen and K.-C. Chou, "Predicting protein fold pattern with functional domain and sequential evolution information," *Journal of Theoretical Biology,* vol. 256, no. 3, pp. 441-446, 2009.

[40] L. Wei, M. Liao, X. Gao and Q. Zou, "Enhanced Protein Fold Prediction Method Through a Novel Feature Extraction Technique," *IEEE Transactions on Nanobioscience,* vol. 14, no. 6, pp. 649-659, 2015.

[41] D. T. Jones, "Protein secondary structure prediction based on position-specific scoring matrices," *Journal of molecular biology,* vol. 292, no. 2, pp. 195-202, 1999.

[42] D. Chen, X. Tian, B. Zhou and J. Gao, "Profold: Protein fold classification with additional structural features and a novel ensemble classifier," *BioMed research international,* 2016.

[43] N. Landwehr, M. Hall and E. Frank, "Logistic model trees," *Machine learning,* vol. 59, pp. 161-205, 2005.

[44] R. A. Sayle and E. J. Milner-White. , "RASMOL: biomolecular graphics for all," *Trends in biochemical sciences,* vol. 20, no. 9, pp. 374-376, 1995.

[45] L. Nanni, "A novel ensemble of classifiers for protein fold recognition," *Neurocomputing,* vol. 69, no. 16, pp. 2434-2437, 2006.

[46] H.-B. Shen and K.-C. Chou, "Ensemble classifier for protein fold pattern recognition," *Bioinformatics,* vol. 22, no. 14, pp. 1717-1722, 2006.

[47] K. Chen and L. Kurgan, "PFRES: protein fold classification by using evolutionary information and predicted secondary structure," *Bioinformatics,* vol. 23, no. 21, pp. 2843-2850, 2007.

[48] T. Yang and V. Kecman, "Adaptive local hyperplane classification," *Neurocomputing,* vol. 71, no. 13, pp. 3001-3004, 2008.

[49] X. Guo and X. Gao, "A novel hierarchical ensemble classifier for protein fold recognition," *Protein Engineering, Design & Selection,* vol. 21, no. 11, pp. 659-664, 2008.

[50] P. Ghanty and N. R. Pal, "Prediction of protein folds: extraction of new features, dimensionality reduction, and fusion of heterogeneous classifiers," *IEEE transactions on nanobioscience,* vol. 8, no. 1, pp. 100-110, 2009.

[51] T. Yang, V. Kecman, L. Cao, C. Zhang and J. Z. Huang, "Margin-based ensemble classifier for protein fold recognition," *Expert Systems with Applications,* vol. 38, no. 10, pp. 12348-12355, 2011.

[52] K. Kavousi, M. Sadeghi, B. Moshiri, B. N. Araabi and A. A. Moosavi-Movahedi, "Evidence theoretic protein fold classification based on the concept of hyperfold," *Mathematical Biosciences,* vol. 240, no. 2, pp. 148-160, 2012.

[53] Z. Feng and X. Hu, "Recognition of 27-class protein folds by adding the interaction of segments and motif information," in *BioMed research international* , 2014.

[54] Z. Feng, X. Hu, Z. Jiang, H. Song and M. A. Ashraf, "The recognition of multi-class protein folds by adding average chemical shifts of secondary structure elements," *Saudi journal of biological sciences,* vol. 23, no. 2, pp. 189-197, 2016.

[55] J. Handl, J. Knowles, R. Vernon, D. Baker and S. C. Lovell, "The dual role of fragments in fragment-assembly methods for de novo protein structure prediction," *Proteins: Structure, Function, and Bioinformatics,* vol. 80, no. 2, pp. 490-504, 2012.

[56] H. Ji, W. Yu and Y. Li., "A rank revealing randomized singular value decomposition (r3svd) algorithm for low-rank matrix approximations," *arXiv preprint arXiv:1605.08134,* 2016.

[57] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing,* vol. 17, no. 4, pp. 395-416, 2007.

[58] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 22, no. 8, pp. 888-905, 2000.

[59] A. Y. Ng, M. I. Jordan and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *In Advances in neural information processing systems,* pp. 849-856, 2002.

[60] N. Halko, P.-G. Martinsson and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review,* vol. 53, no. 2, pp. 217-288, 2011.

[61] Y. Gu, W. Yu and Y. Li, "Efficient randomized algorithms for adaptive low-rank factorizations of large matrices," *arXiv preprint arXiv:1606.09402,* 2016.

[62] Y. Li and W. Yu, "A Fast Implementation of Singular Value Thresholding Algorithm using Recycling Rank Revealing Randomized Singular Value Decomposition," *arXiv preprint arXiv:1704.05528,* 2017.

[63] C. Strobl, A.-L. Boulesteix and A. Zeil, "Bias in random forest variable importance measures: Illustrations, sources and a solution," , vol. 8, Jan 25 .," *Bmc Bioinformatics,* vol. 8, no. 1, p. 25, 2007.

[64] G. Wang and . R. L. Dunbrack Jr., "PISCES: a protein sequence culling server," *Bioinformatics,* vol. 19, no. 12, pp. 1589-1591, 2003.

[65] R. Kolodny, P. Koehl, L. Guibas and M. Levitt, "Small libraries of protein fragments model native protein structures accurately," *Journal of molecular biology,* vol. 323, no. 2, pp. 297-307, 2002.

[66] Y. Chiang, T. Gelfand, A. E. Kister and I. M. Gelfand, "New classification of supersecondary structures of sandwich-like proteins uncovers strict patterns of strand assemblage," *Proteins: Structure, Function, and Bioinformatics,* vol. 68, no. 4, pp. 915-921, 2007.

[67] K. T. Simons, I. Ruczinski, C. Kooperberg, B. A. Fox and D. Baker, "Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins," *Proteins: Structure, Function, and Bioinformatics,* vol. 34, no. 1, pp. 82-95, 1999.

[68] Y. Li, "Conformational sampling in template-free protein loop structure modeling: an overview," *Computational and structural biotechnology journal,* vol. 5, no. 6, p. e201302003, 2013.

[69] A. Bazzoli, A. G. Tettamanzi and Y. Zhang, "Computational protein design and large-scale assessment by I-TASSER structure assembly simulations," *Journal of molecular biology,* vol. 407, no. 5, pp. 764-776, 2011.

[70] W. Elhefnawy, L. Chen, Y. Han and Y. Li, "ICOSA: A distance-dependent, orientation-specific coarse-grained contact potential for protein structure modeling," *Journal of molecular biology,* vol. 427, no. 15, pp. 2562-2576, 2015.

[71] W. Elhefnawy, M. Li, J. Wang and Y. Li, "Construction of protein backbone fragments libraries on large protein sets using a randomized spectral clustering algorithm," in *International symposium on bioinformatics research and applications*, 2017.

[72] W. Elhefnawy, M. Li, J.-X. Wang and Y. Li, "Decoding the Structural Keywords in Protein Structure Universe," *Journal of Computer Science and Technology,* vol. 34, no. 1, pp. 3-15, 2019.

[73] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," *In Advances in neural information processing systems,* pp. 2222-2230, 2012.

[74] S. Min, B. Lee and S. Yoon, "Deep learning in bioinformatics," *Briefings in bioinformatics,* vol. 18, no. 5, pp. 851-869, 2017.

[75] Y.-d. Cai and S. L. Lin, "Support vector machines for predicting rRNA-, RNA-, and DNA-binding proteins from amino acid sequence," *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics,* vol. 1648, no. 1-2, pp. 127-133, 2003.

[76] P. Mundra, M. Kumar, K. K. Kumar, V. K. Jayaraman and B. D. Kulkarni, "Using pseudo amino acid composition to predict protein subnuclear localization: Approached with PSSM," *Pattern Recognition Letters,* vol. 28, no. 13, pp. 1610-1615, 2007.

[77] D. T.-H. Chang, H.-Y. Huang, Y.-T. Syu and C.-P. Wu, "Real value prediction of protein solvent accessibility using enhanced PSSM features," *BMC bioinformatics,* vol. 9, no. 12, p. S12, 2008.

[78] J. cheol Jeong, X. Lin and X.-w. Chen, "On position-specific scoring matrix for protein function prediction," *IEEE/ACM transactions on computational biology and bioinformatics,* vol. 8, no. 2, pp. 308-315, 2011.

[79] Y. Ohta, Y. Ogura and A. Wada, "Thermostable Protease from Thermophilic Bacteria I. THERMOSTABILITY, PHYSICOCHEMICAL PROPERTIES, AND AMINO ACID COMPOSITION," *Journal of Biological Chemistry,* vol. 241, no. 24, pp. 5919-5925, 1966.

[80] I. Goodfellow, Y. Bengio, A. Courville and Y. Bengio, Deep learning, vol. 1, Cambridge: MIT press, 2016.

[81] J. Zhu, H. Zhang, S. Cheng Li, C. Wang, L. Kong, S. Sun, W.-M. Zheng and D. Bu, "Improving protein fold recognition by extracting fold-specific features from predicted residue–residue contacts," *Bioinformatics,* vol. 33, no. 23, pp. 3749-3757, 2017.

[82] W.-Y. Yang, B.-L. Lu and Y. Yang, "A comparative study on feature extraction from protein sequences for subcellular localization prediction," in *2006 IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology*, 2006.

[83] M. O. Dayhoff, R. M. Schwartz and B. C. Orcutt., "22 a model of evolutionary change in proteins," *Atlas of protein sequence and structure,* pp. 345-352, 1978.

[84] B. J. Strait and T. G. Dewey, "The Shannon information entropy of protein sequences," *Biophysical journal,* vol. 71, no. 1, p. 148, 1996.

[85] I. Dubchak, I. Muchnik , S. R. Holbrook and S.-H. Kim, "Prediction of protein folding class using global description of amino acid sequence," in *Proceedings of the National Academy of Sciences 92*, 1995.

[86] H.-B. Shen and K.-C. Chou, "PseAAC: a flexible web server for generating various kinds of protein pseudo amino acid composition," *Analytical biochemistry,* vol. 373, no. 2, pp. 386-388, 2008.

[87] H. Saini, G. Raicar, A. Sharma, S. Lal, A. Dehzangi, J. Lyons, K. K. Paliwal, S. Imoto and S. Miyano, "Probabilistic expression of spatially varied amino acid dimers into general form of Chou ′s pseudo amino acid composition for protein fold recognition," *Journal of theoretical biology ,* vol. 380, pp. 291-298, 2015.

[88] P. Lio, "Wavelets in bioinformatics and computational biology: state of art and perspectives," *Bioinformatics 19*, no. 1, pp. 2-9, 2003.

[89] T. Liu, X. Zheng and J. Wang, "Prediction of protein structural class for low-similarity sequences using support vector machine and PSI-BLAST profile," *Biochimie,* vol. 92, no. 10, pp. 1330-1334, 2010.

[90] A. Dehzangi, S. Phon-Amnuaisuk and O. Dehzangi, "Using Random Forest for Protein Fold Prediction Problem: An Empirical Study," *J. Inf. Sci. Eng.,* vol. 26, no. 6, pp. 1941-1956, 2010.

[91] A. Dehzangi, S. Phon-Amnuaisuk, M. Manafi and S. Safa, "Using Rotation Forest for Protein Fold Prediction Problem: An Empirical Study," in *European Conference on Evolutionary*

Computation, Machine Learning and Data Mining in Bioinformatics, Berlin, Heidelberg, 2010.

[92] K. K. Paliwal, A. Sharma, J. Lyons and A. Dehzangi, "A tri-gram based feature extraction technique using linear probabilities of position specific scoring matrix for protein fold recognition," *IEEE transactions on nanobioscience,* vol. 13, no. 1, pp. 44-50, 2014.

[93] K. K. Paliwal, A. Sharma, J. Lyons and A. Dehzangi, "Improving protein fold recognition using the amalgamation of evolutionary-based and structural based information," *BMC bioinformatics*, vol. 15, no. 16, p. S12, 2014.

[94] A. Dehzangi, K. Paliwal, J. Lyons, A. Sharma and A. Sattar, "A segmentation-based method to extract structural and evolutionary features for protein fold recognition," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB),* vol. 11, no. 3, pp. 510-519, 2014.

[95] J. Lyons, A. D. Dehzangi, R. Heffernan, Y. Yang, Y. Zhou, A. Sharma and K. Paliwal, "Advancing the Accuracy of Protein Fold Recognition by Utilizing Profiles from Hidden Markov Models," *IEEE Transactions on Nanobioscience,* vol. 14, no. 7, pp. 761-772, 2015.

[96] J. Lyons, K. K. Paliwal, A. Dehzangi, R. Heffernan, T. Tsunoda and A. Sharma, "Protein fold recognition using HMM–HMM alignment and dynamic programming," *Journal of theoretical biology*, vol. 393, pp. 67-74, 2016.

[97] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, "Object detectors emerge in deep scene cnns," *arXiv preprint arXiv:1412.6856,* 2014.

[98] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, 2014.

[99] H.-X. Zhou and X. Pang, "Electrostatic interactions in protein structure, folding, binding, and condensation," *Chemical reviews*, vol. 118, no. 4, pp. 1691-1741, 2018.

APPENDIX I

- **DD Protein Fold Dataset**

Table 8

DD dataset folds from SCOP.

Index	Fold ID	Fold Name	Training	Testing	Total
1	a.1	Globin-like	13	6	19
2	a.3	Cytochrome c	7	9	16
3	a.4	DNA/RNA-binding 3-helical bundle	12	30	32
4	a.24	4-Helical up-and-down bundle	7	8	15
5	a.26	4-Helical cytokines	9	9	18
6	a.39	EF hand-like	6	9	15
7	b.1	Immunoglobulin-like β-sandwich	30	44	74
8	b.6	Cupredoxin-like	9	12	21
9	b.121	Nucleoplasmin-like/VP	16	13	29
10	b.29	ConA-like lectins/glucanases	7	6	13
11	b.34	SH3-like barrel	8	8	16
12	b.40	OB-Fold	13	19	32
13	b.42	β-Trefoil	8	4	12
14	b.47	Trypsin-like serine proteases	9	4	13
15	b.60	Lipocalins	9	7	16
16	c.1	TIM β/α-barrel	29	48	77
17	c.2	FAD/NAD(P)-binding domain	11	12	23
18	c.3	Flavodoxin-like	11	13	24
19	c.23	NAD(P)-binding Rossmann	13	27	40
20	c.37	P-loop containing NTH	10	12	22
21	c.47	Thioredoxin-fold	9	8	17
22	c.55	Ribonuclease H-like motif	10	12	22
23	c.69	α/β-Hydrolases	11	7	18

24	c.93	Periplasmic binding protein-like	11	4	15
25	d.15	β-Grasp (ubiquitin-like)	7	8	15
26	d.58	Ferredoxin-like	13	27	40
27	g.3	Knottins (small inhibitors, toxins, lectins)	13	27	40
Total			311	383	694

- **EED Protein Fold Dataset**

Table 9

EED dataset folds from SCOP

Index	Fold ID	Fold Name	# of Samples
1	a.1	Globin-like	41
2	a.3	Cytochrome c	35
3	a.4	DNA/RNA-binding 3-helical bundle	322
4	a.24	4-Helical up-and-down bundle	69
5	a.26	4-Helical cytokines	30
6	a.39	EF hand-like	59
7	b.1	Immunoglobulin-like β-sandwich	391
8	b.6	Cupredoxin-like	47
9	b.121	Nucleoplasmin-like/VP	60
10	b.29	ConA-like lectins/glucanases	57
11	b.34	SH3-like barrel	129
12	b.40	OB-Fold	156
13	b.42	β-Trefoil	45
14	b.47	Trypsin-like serine proteases	45
15	b.60	Lipocalins	37

16	c.1	TIM β/α-barrel	336
17	c.2	FAD/NAD(P)-binding domain	73
18	c.3	Flavodoxin-like	130
19	c.23	NAD(P)-binding Rossmann	195
20	c.37	P-loop containing NTH	239
21	c.47	Thioredoxin-fold	111
22	c.55	Ribonuclease H-like motif	128
23	c.69	α/β-Hydrolases	83
24	c.93	Periplasmic binding protein-like	16
25	d.15	β-Grasp (ubiquitin-like)	121
26	d.58	Ferredoxin-like	339
27	g.3	Knottins (small inhibitors, toxins, lectins)	124

- **TG Protein Fold Dataset**

Table 10

TG-dataset.

Index	Fold ID	Fold Name
1	a.3	Cytochrome C
2	a.4	DNA/RNA binding 3-helical bundle
3	a.24	Four helical up and down bundle
4	a.39	EF hand-like fold
5	a.60	SAMdomain-like

6	a.118	a-a superhelix
7	b.1	Immunoglobulin-like b-sandwich
8	b.2	Common fold of diphtheria toxin/transcription factors/cytochrome f
9	b.6	Cupredoxin-like
10	b.18	Galactose-binding domain-like
11	b.29	Concanavalin A-like lectins/glucanases
12	b.34	SH3-like barrel
13	b.40	OB-fold
14	b.82	Double-stranded a-helix
15	b.121	Nucleoplasmin-like
16	c.1	TIM a/b-barrel
17	c.2	NAD(P)-binding Rossmann-fold domains
18	c.3	FAD/NAD(P)-binding domain
19	c.23	lavodoxin-like
20	c.26	Adenine nucleotide a hydrolase-like
21	c.37	P-loop containing nucleoside triphosphate hydrolases
22	c.47	Thioredoxin fold
23	c.55	Ribonuclease H-like motif
24	c.66	S-adenosyl-L-methionine-dependent methyltransferases
25	c.69	a/b-Hydrolases

26	d.15	b-Grasp, ubiquitin-like
27	d.17	Cystatin-like
28	d.58	Ferredoxin-like
29	g.3	Knottins
30	g.41	Rubredoxin-like

- **SCOP 2.04 TOP 40 Folds**

Table 11

SCOP 2.04 top 40 folds

Fold	Class	Description	# Proteins
b.1	b: All beta proteins	Immunoglobulin-like beta-sandwich	529
c.1	c: Alpha and beta proteins (a/b)	TIM beta/alpha-barrel	485
d.58	d: Alpha and beta proteins (a+b)	Ferredoxin-like	424
a.4	a: All alpha proteins	DNA/RNA-binding 3-helical bundle	387
c.2	c: Alpha and beta proteins (a/b)	NAD(P)-binding Rossmann-fold domains	309
c.37	c: Alpha and beta proteins (a/b)	P-loop containing nucleoside triphosphate hydrolases	307
c.23	c: Alpha and beta proteins (a/b)	Flavodoxin-like	216

c.47	c: Alpha and beta proteins (a/b)	Thioredoxin fold	195
b.40	b: All beta proteins	OB-fold	174
b.34	b: All beta proteins	SH3-like barrel	170
c.55	c: Alpha and beta proteins (a/b)	Ribonuclease H-like motif	159
c.94	c: Alpha and beta proteins (a/b)	Periplasmic binding protein-like II	154
a.118	a: All alpha proteins	alpha-alpha superhelix	146
d.15	d: Alpha and beta proteins (a+b)	beta-Grasp (ubiquitin-like)	144
c.66	c: Alpha and beta proteins (a/b)	S-adenosyl-L-methionine-dependent methyltransferases	140
g.3	g: Small proteins	Knottins (small inhibitors, toxins, lectins)	138
b.82	b: All beta proteins	Double-stranded beta-helix	126
c.69	c: Alpha and beta proteins (a/b)	alpha/beta-Hydrolases	121
c.67	c: Alpha and beta proteins (a/b)	PLP-dependent transferase-like	118
d.17	d: Alpha and beta proteins (a+b)	Cystatin-like	103

d.144	d: Alpha and beta proteins (a+b)	Protein kinase-like (PK-like)	100
c.108	c: Alpha and beta proteins (a/b)	HAD-like	99
d.108	d: Alpha and beta proteins (a+b)	Acyl-CoA N-acyltransferases (Nat)	92
c.26	c: Alpha and beta proteins (a/b)	Adenine nucleotide alpha hydrolase-like	91
a.60	a: All alpha proteins	SAM domain-like	87
b.29	b: All beta proteins	Concanavalin A-like lectins/glucanases	86
b.36	b: All beta proteins	PDZ domain-like	84
b.55	b: All beta proteins	PH domain-like barrel	84
c.93	c: Alpha and beta proteins (a/b)	Periplasmic binding protein-like I	84
a.39	a: All alpha proteins	EF Hand-like	80
a.24	a: All alpha proteins	BSD domain-like	77
c.3	c: Alpha and beta proteins (a/b)	FAD/NAD(P)-binding domain	74
d.38	d: Alpha and beta proteins (a+b)	Thioesterase/thiol ester dehydrase-isomerase	72
g.37	g: Small proteins	beta-beta-alpha zinc fingers	71
a.25	a: All alpha proteins	Ferritin-like	68

b.18	b: All beta proteins	Galactose-binding domain-like	68
b.121	b: All beta proteins	Nucleoplasmin-like/VP (viral coat and capsid proteins)	68
g.39	g: Small proteins	Glucocorticoid receptor-like (DNA-binding domain)	67
c.56	c: Alpha and beta proteins (a/b)	Phosphorylase/hydrolase-like	66
c.14	c: Alpha and beta proteins (a/b)	TTHA0583/YokD-like	64
d.129	d: Alpha and beta proteins (a+b)	TBP-like	63
a.29	a: All alpha proteins	Bromodomain-like	62
d.110	d: Alpha and beta proteins (a+b)	Profilin-like	62

VITA

Wessam Elhefnawy

Department of Computer Science

Old Dominion University

Norfolk, VA 23529

Wessam received his Bachelor and Master degrees in Computer Engineering from the Arab Academy for Science and Technology, Cairo, Egypt, in 2004 and 2011, respectively. In Fall 2013, he joined the Computer Science Department of Old Dominion University and started his research in computational biology and machine learning. Wessam's research objectives are directed toward studying and implementing novel computational biology and machine learning algorithms to accommodate biological and chemical experiments on proteins. He developed several computational methods for a set of fundamental and universal bioinformatics challenges, such as identifying conformational clusters of phosphorylated tyrosine, creating protein fragment libraries, and designing and implementing a protein fold recognition model. In addition to the academic work, Wessam joined Intel innovation technology for a six months machine learning internship in 2018.