Protein Structural Class and Ligand Binding Prediction Using Image Based Feature

Al Amin Neaz Ahmed
Student Id: 011 141 147
Nafees Sadique
Student Id: 011 151 244
MD Tajul Islam
Student Id: 011 142 112
Md. Nawshad Pervage
Student Id: 011 142 118

United International University

Dhaka, Bangladesh

March 2019

This thesis was submitted for the degree of
BSc in Computer Science & Engineering

Declaration

We, [Al Amin Neaz Ahmed, Nafees Sadique, Md Tajul Islam and Md. Nawshad Pervage], declare that this thesis titled, Thesis Title and the work presented in it are our own. I confirm that:

- This work was done wholly or mainly while in candidature for a [BSc] degree at United International University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at United International University or any other institution, this has been clearly stated.
- Where we have consulted the published work of others, this is always clearly attributed.
- Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely our own work.
- We have acknowledged all main sources of help.
- Where the thesis is based on work done by ourselves jointly with others, We have made clear exactly what was done by others and what we have contributed ourselves.

____

[Al Amin Neaz Ahmed, 011 141 147, Computer Science & Engineering]

____

[Nafees Sadique, 011 151 244, Computer Science & Engineering]

____

[Md Tajul Islam, 011 142 112, Computer Science & Engineering]

____

[Md. Nawshad Pervage, 011 142 118, Computer Science & Engineering]

Certificate

I do hereby declare that the research works embodied in this thesis entitled “**Protein Structural Class and Ligand Binding Prediction Using Image Based Feature**” is the outcome of an original work carried out by [Al Amin Neaz Ahmed, Nafees Sadique, Md Tajul Islam and Md. Nawshad Pervage] under my supervision.

I further certify that the dissertation meets the requirements and the standard for the degree of [MSc/ BSc] in Computer Science and Engineering.

[Dr. Swakkhar Shatabda, Associate Professor & Undergraduate Program Coordinator]

Abstract

Proteins are the building blocks of all cells in both human and all our living creatures of the world. Most of the work in the living organism is performed by Proteins. Proteins are polymers of amino acid monomers which are biomolecules or macromolecules. The tertiary structure of protein represents the three-dimensional shape of a protein. The functions, classification and binding sites are governed by protein's tertiary structure. If two protein structures are alike then the two proteins can be of the same kind. To detect the similarity of proteins accurately in real time is crucial in the research.

In this thesis, we present an analysis on local binary pattern histogram, Wavelet transformed Local Binary Pattern Histogram, Separate Row Multiplication Matrix with Uniform Local Binary Pattern Histogram, Neighbor Block Subtraction Matrix with Uniform Local Binary Pattern Histogram and Atom Bond for protein structural class prediction. We have used them on the distance matrix of α carbons of proteins which are used as an image for feature extraction.

The experiments were done on a 40 percent reduced dataset of PDB files. We have demonstrated the usefulness of this feature over a large variety of supervised machine learning algorithms. We propose the use of Random Forest as the best performing classifier on this dataset using the selected features.

Secondly, Protein-Ligand binding is accountable for managing the tasks of biological receptors that helps to cure diseases and many more. So, binding prediction between protein and ligand is important for understanding a protein's activity or to accelerate docking computations in virtual screening-based drug design.

Protein-Ligand Binding Prediction requires three-dimensional tertiary structure of the target protein to be searched for ligand binding. In this paper, we've introduced a supervised learning algorithm for predicting Protein-Ligand Binding which is a Similarity-Based Clustering approach.

Our algorithm works better than most popular and widely used machine learning algorithms.

So, our work is divided into two parts, Protein Structural Class Prediction & Protein-Ligand Binding Prediction.

Acknowledgement

We thank Rezaul Karim for sharing the SQL dataset files and algorithms to generate Distance Matrix from PDB files for CoMOGrad and PHOG and Md Mosharof Hossain for lending his personal computer.

Table of Contents

LIST OF TABLES viii

LIST OF FIGURES ix

1. Introduction 1

1.1 Protein Structural Class Prediction 1

1.2 Protein-Ligand Binding Prediction 3

2. Background and Literature Review 4

2.1 Biological background 4

2.1.1 Protein 4

2.1.2 Ligand 4

2.1.3 Protein Structure 5

2.2 Protein Structural Class Prediction 5

2.3 Protein-Ligand Binding Prediction 7

2.2.1 Background 7

2.2.2 Related Work 8

3. Methodology 10

3.1 Protein Structural Class Prediction 10

3.1.1 Dataset 11

3.1.2 Image Generation 11

3.1.3 Scaling Images to Same Dimension 11

3.1.4 Feature Extraction 12

3.1.5 Re-evaluate Dataset 15

3.1.6 Removing Multiclass Imbalance Problem 15

3.1.7 Classifiers 16

3.1.8 Performance Evaluation 16

3.2 Protein-Ligand Binding Prediction 17

3.2.1 Dataset 17

3.2.2 Data pre-processing 17

3.2.3 Our Classifier 19

4. Results 23

4.1 Protein Structural Class Prediction 23

4.1.1 Parameters used for the classifiers 23

4.1.2 Analysis of Features 23

4.1.3 Comparison with other methods 24

4.1.4 Discussion 25

4.2 Protein-Ligand Binding Prediction 26

4.2.1 Dependencies/Hyperparameters 26

4.2.2 Classifiers 26

4.2.3 Comparison with other Classifiers 27

5. Conclusion 30

5.1 Protein Structural Class Prediction 30

5.2 Protein-Ligand Binding Prediction 30

6. References 31

LIST OF TABLES

Table 1. Protein Classes and its Corresponding Instances 11

Table 2. Feature Group For Protein Class Prediction..15

Table 3. Accuracy of different classifiers of protein images...24

Table 4. Comparison of the method proposed features in this paper with [8] and [11] 25

Table 5. Accuracy Comparison Table 28

Table 6. Sensitivity Comparison Table 28

LIST OF FIGURES

Figure 1. Peptide Bond 4

Figure 2. Block diagram of Protein Structural Class Prediction methodology 10

Figure 3. An example of basic LBP.. 12

Figure 4. An example of Separate Row Multiplication Matrix with Uniform Local Binary Pattern Histogram 14

Figure 5. Clustering-Based Undersampling 18

Figure 6. Block Diagram of Similarity Based Clustering for Protein-Ligand Binding Prediction.. 19

Figure 7. Relation between given Protein and Related Ligands.. 21

Figure 8. Relation between given Ligand and Related Proteins 21

Chapter1

Introduction

In this thesis we've done two related works, finding the novel features for Protein Class Prediction & proposing a new algorithm for Protein-Ligand Binding Prediction. Each of them is presented separately in different sections in each Chapter. Rest of the paper is organized as following: Chapter 2 briefly presents a literature review of the related work; Chapter 3 describes the methodology and materials proposed in this paper; experimental results are shown in Chapter 4 with a discussion and the paper conclude in Chapter 5.

1.1 Protein Structural Class Prediction

Protein tertiary structure comparison is very important in many applications of modern structural biology, drug design, drug discovery, in studies of protein-protein interactions and other fields. This is especially significant because the structure of a protein is more protected than the protein sequence [1]. Many works have been done to find protein binding [2].

Comparison of protein structure has been done in many works of literature by alignment of distance matrices [3], using iterated double dynamic programming [4], using elastic shape analysis [5] and many other techniques. The most common way of comparing protein tertiary structure is to treat the protein as a three-dimensional object and superimpose one on another. Different distances are used to calculate the differences between the proteins.

The distance matrix of α carbon can be seen extensively used in [6] [7] as a feature which represents the tertiary structure of a protein chain. This feature is used as a feature vector which represents the structure of a protein to measure either similarity or dissimilarity to measure and compare the feature vectors with one another in pattern recognition literature. A mapped two-dimensional feature matrix is created from the 3D coordinate data of protein. The intra-molecular distance is used to make the α carbon distance matrix which mirrors the tertiary structure of a protein and the conserved elements of the secondary structure in it. With an input matrix size of N x N, the distance matrix based exact algorithms run in 0(N!) time [8].

An image is basically a matrix of N x N dimension with corresponding data in each cell. Thus, the distance matrix can be used as an image. Basically, three types of features can be generated from an image: pixel based, filter based and computationally generated features. Pixel based features e.g. histograms are simplistic and dependent on the capability of classification algorithms. Filter based methodologies transform the original image to use feature extraction methods. Refined algorithms are used to segment and other various algorithms are used to detect different features.

Using ideas from computer vision and utilizing it in protein structure retrieval is not uncommon in the field. ProteinDBS server [9] implement a similar approach in [10] by Chietal. Texture features from the original size images and diagonally partitioned images were extracted by Chi et al. CoMOGrad and PHOG [8] also used images to extract their two novel features whereas we are extracting histograms of local binary pattern images from the original image.

In this paper, we propose the combination of local binary pattern histogram, Wavelet transformed Local Binary Pattern Histogram, Separate Row Multiplication Matrix with Uniform Local Binary Pattern Histogram, Neighbor Block Subtraction Matrix with Uniform Local Binary Pattern Histogram and Atom Bond features to be used for protein similarity measurement. We extract the distance matrix of α carbon of a protein from PDB file and use the distance matrix as an image to extract our first four features and Atom Bond is extracted from the PDB files. We have used a large variety of classification algorithms to test the extracted features. We are also going to show the results and comparative study of different implementation methodologies such as wavelet and pyramid histogram-based features [11] and CoMOGrad and PHOG. The method we have proposed is able to produce a better result on some classification algorithm over the previous methods on the same benchmark.

1.2 Protein-Ligand Binding Prediction

Human body uses protein for repairing tissues, making enzymes, hormones, and other biological chemicals. It is an essential building block of bones, muscles, cartilage, skin, and blood. On the other hand, a ligand is a material that has the potentiality to bind to and forms a composite with a biomolecule in order to carry out a biological function. In Protein-Ligand Binding, the ligand is usually a molecule which produces a signal by binding to a locus on a target protein. The binding typically results in a change of conformational isomerism (conformation) of the target protein. The evolution of the protein's responsibility depends on the development of specific sites which are designed to bind ligand molecules. Ligand binding ability is important for the management of biological functions. Ligand binding interactions changes the protein state and function. Protein-Ligand Binding prediction is very important in many applications of modern structural biology, drug design, drug discovery and other fields.

We can compare Protein-Ligand interactions with lock and key approach. Let's assume protein as a lock and ligand as a key. So, for interactions their binding space need to be perfectly matched. Based on the tertiary structure of both protein and ligand, binding between them can be predicted using x, y, z coordinates of the atoms of the proteins and ligands. We've introduced Similarity-Based Clustering method for the Binding prediction as we have supervised data. So, the higher the number of supervised training data is, the higher the chance of accurate prediction is.

Our algorithm is a combination of KNN [25] and clustering methodology. Where traditional machine learning algorithms performs poor than random classification, our algorithms works better than those.

Chapter 2

Background and Literature Review

2.1 Biological background

2.1.1 Protein

Protein is a large biomolecule or macromolecule consisting of one or more long chains of amino acid. 50% of the Cellular Dry Weight is protein. Humans have about 25,000 genes. About 20,000 of these genes are protein-coding genes. That means humans make at least 20,000 proteins. Not all of them are different since the number of protein-coding genes includes many duplicated genes and gene families. There are 300 amino acids and only 20 of them occur in protein.

Multiple amino acid makes Peptide Bond (Figure 1) between Amine (NH_2) and Carboxyl (-COOH) group to produce a chain that represents a protein and releases water (H_2O).

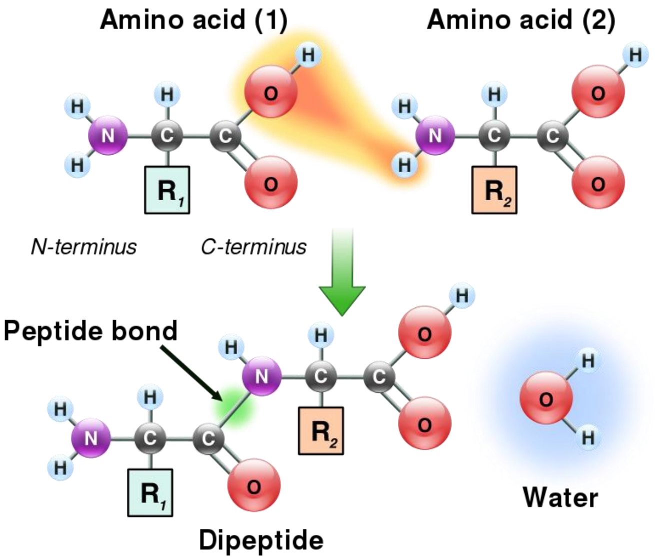


Figure 1. Peptide Bond

2.1.2 Ligand

In coordination chemistry, a ligand is an ion or molecule (functional group) that binds to a central metal atom to form a coordination complex. The bonding with the metal generally involves formal donation of one or more of the ligand's electron pairs. The nature of metal–ligand bonding can range from covalent to ionic. Furthermore, the metal–ligand bond order can range from one to three. Ligands are viewed as Lewis bases, although rare cases are known to involve Lewis acidic "ligands".

2.1.3 Protein Structure

3 types of protein structures are there.

i. Primary Structure: It is a sequence of amino acids present in polypeptide chain. Each amino acid in this chain is “residue”. Amino Acids have covalent bonds only.

ii. Secondary Structure: In this structure, residues has non-covalent bonds where maximum of the bonds are Hydrogen-Bonds. Two types of Secondary structures are

   a. α-helix: It is a right-handed spiral structure and tightly packed. The peptide bonds work as the backbone core. Side chains extends outwards. This structure is stabilized by Hydrogen bonding between carbonyl oxygen and amide hydrogen. Number of amino acids per turn is 3.6 angstrom and vertical distance between consecutive tuns of the helix is 5.4 angstrom.

   b. β-sheet: This is the structure when two or more polypeptides line up side by side. Individual polypeptide is called β-strand. Each of them is fully extended. This structure is stabilized by Hydrogen bond between NH^+ and COO- groups of adjacent chains. Side by side polypeptides an be parallel or anti-parallel.

iii. Tertiary Structure: It is a3-D structure based on various types of interactions between the side chains of the peptide chain. The α-helixes and β-pleated-sheets are folded into a compact globular structure. The structure is stable only when the parts of a protein domain are locked into place by specific tertiary interactions, such as salt bridges, hydrogen bonds, and the tight packing of side chains and disulfide bonds. The atomic coordinates of most of these structures are deposited in a database known as the protein data bank (PDB). It allows the tertiary structures of a variety of proteins to be analyzed and compared.

2.2 Protein Structural Class Prediction

There are experiments performed to compare protein structure as three-dimensional objects. Score function based on different distance metrics to find similarity and dissimilarity as a measurement has been proposed by these methods. The most prominent improvement of the literature is presented briefly below.

A. *DALI*

Distance Alignment Matrix Method (DALI) [6] calculates an alignment score by finding an absolute alignment between the α carbon distance matrices of proteins. A distance matrix is created by breaking the input structure into hexapeptide fragments and evaluating the contact patterns(pair-wise) between them and making a list with a matching score by saving the matching pairs [3]. The final matching score and overall alignment are made by gathering pairs in the correct order. Monte Carlo optimization is used for the assembling.

B. *CE*

Combinatorial Extension (CE) [12] is comparable to DALI because it creates a series of fragments by breaking each structure in the query set and later attempts to reassemble toward a complete alignment. Protein structures are compared by using combinatorial extension and Monte Carlo optimization. The computational cost is quite huge to implement this method despite having good accuracy. Thus, a real-time web service cannot be implemented due to its cost ineffectiveness.

C. *SSAP*

The Sequential Structure Alignment Program (SSAP) [13] uses β carbons unlike the other methods using α carbon of protein in structural alignment except for glycine. Double Dynamic programming is used to produce atom-to-atom vectors which is based on structural alignment in structure space. Inter-residue distance vectors amid every individual residue and its nearest non-contiguous neighbors on each protein are first generated. The vector differences amid neighbors are created in a series of matrices. Optimal local alignments are found from each resulting matrix by applying dynamic programming. A 'summary' matrix is created from the summed up optimal local alignment. A comprehensive structural alignment is resolved by applying dynamic programming again.

D. *FATCAT*

Flexible structure AlignmenT by Chaining Aligned fragment pairs with Twists (FATCAT)[14] treats the protein structure like a fixed body. It produces good results for maximum cases with other fixed body approaches [15].

E. *ProteinDBS*

ProteinDBS [9] compares α carbon distance matrix images by using some common features of CBIR (Content Based Image Retrieval). It correlates only some particular image features thus it performs much faster than the previous ones. The drawback of ProteinDBS is the expensive cost of computation.

F. *TM-align and SP-align*

The most well-known method for protein structure alignment is TM-align [16] [17]. Finding an optimal alignment and an alignment matrix it computes the TM-Score. SP-Align [18] is also a popular approach which is similar to TM-align. The difference between the two lies in the alignment algorithm and the alignment score.

G. *CoMOGrad and PHOG*

CoMOGrad stands for Co-occurrence Matrix of the Oriented Gradient of Distance Matrices and PHOG stands for Pyramid Histogram of Oriented Gradient [8]. This methodology also uses the α carbon distance matrix of protein. The dimension of all distance matrix is converted to 128 × 128. In CoMOGrad, the gradient angle and magnitude is computed from the distance matrix and the values are quantized. Quantization is a compressing technique which compresses a range of values to a single quantum value. In this methodology, the values are quantized to 16 bins which produce a co-occurrence matrix which is 16 × 16 matrix. The matrix is converted into a vector of size 256. Quadtree from the distance matrix is created with the desired level in PHOG. Gradient Oriented Histogram of each node is calculated with the preferred number of bins and bin size. In gradient oriented histogram an image is divided into small sub-images called cells and histogram of edge orientations are accumulated within the cell. The combined histogram entries are used as the feature vector describing the object. Total features which are the multiplication of total nodes and number of bins are incorporated in the vector with the size of the total number of features. The vector is normalized by dividing it with the sum of its components.

2.3 Protein-Ligand Binding Prediction

2.2.1 Background

In order to understand the related works and before diving into our methodology, we first need to have some background knowledge that would be helpful for understanding the methodologies more efficiently. These points are discussed below.

A. Tertiary Protein Structure

Protein tertiary structure is defined by its atomic coordinates which refers to the three-dimensional shape of protein. Protein is a chain of amino acid where each of them has alpha carbon. Coordinates of these alpha carbon defines the tertiary structure precisely. In a protein PDB, 3D coordinates of the atoms are given sequentially which is gained from tertiary structure.

B. Ligand Binding

Ligand is a substance that forms a complex with a biomolecule to perform a biological task. Usually it is a small sized biological element having few atoms. In protein-ligand binding, the ligand is usually a molecule which produces a signal by binding to a site on a target protein. The structure of a protein, for example an enzyme, may change upon binding of its natural ligands. So, learning ligand binding is essential for predicting the functions of the biological components.

C. Clustering

Clustering is the task of grouping objects based on their similarity. It is commonly used in machine learning, data mining, pattern recognition, image analysis, bioinformatics, computer graphics etc. Here, similarity refers to distance between objects. Small distance represents higher similarity. Every cluster has a cluster center. Cluster center can be the average of the objects within the cluster, average of minimum and maximum distance between the objects.

2.2.2 Related Work

Many experimental techniques can be used to investigate various aspects of protein–ligand binding. X-ray crystallography, nuclear magnetic resonance(NMR), Laue X-ray diffraction, small-angle X-ray scattering, and cryo-electron microscopy provide atomic-resolution or near-atomic-resolution structures of the unbound proteins and the protein–ligand complexes, which can be used to study the changes in structure and/or dynamics between the free and bound forms as well as relevant binding events.

Although experimental techniques can investigate thermodynamic profiles for a ligand–protein complex, the experimental procedures for determination of binding affinity are laborious, time-consuming, and expensive. Modern rational drug design usually involves the HTS of a large compound library comprising hundreds or thousands of compounds to find the lead molecules, but this is still not realistic using experimental methods alone.

Some of both experimental and theoretical methods are presented briefly below.

A. Isothermal Titration Calorimetry (ITC)

The structural and dynamic data alone, even when coupled with the most sophisticated computational methods, cannot provide information about the complete thermodynamic profiles consisting of the binding free energy, entropy, and enthalpy, and, therefore, may not accurately predict the binding affinity [29].

B. *Surface Plasmon Resonance (SPR)*

SPR [30], which is an optical-based method to measure the change in the refractive index near a sensor surface, is label-free and capable of measuring real-time quantification of protein–ligand binding kinetics and affinities.

C. *Fluorescence Polarization (FP)*

Fluorescence has a wide spectrum of wavelengths, and, therefore, multiple colors can be applied for detecting the binding of the fluorescent-labelled ligand to a target. Fluorescence-based techniques [31] used for investigating intermolecular interactions.

D. *Protein–Ligand Docking*

Protein–ligand docking [32], which is a branch of the molecular docking field, represents a particularly important methodology due to its importance in the current drug discovery process. Protein–ligand docking methods contain two essential components: the search algorithm and the scoring function. The former is responsible for searching through different ligand conformations and orientations (poses) within a given target protein; the latter is responsible for estimating the binding affinities of the generated poses, ranking them, and identifying the most favorable binding mode(s) of the ligand to the given target.

E. *Free Energy Calculations*

Free energy calculations [33] of the protein-ligand binding try to compute the binding free energies based on the principles of statistical thermodynamics. Such calculations are commonly based on extensive computational simulations (i.e., MD or MC) of the protein and ligand and, as such, require computational efforts several orders of magnitude higher than the traditional scoring functions.

Chapter 3

Methodology

3.1Protein Structural Class Prediction

In this section, we are going to describe our methodology. Atom bond features are generated from the PDB files. Images are created from the α carbon of protein collected from the PDB files of the given dataset. Separate Row Multiplication Matrix with Uniform Local Binary Pattern Histogram, Neighbor Block Subtraction Matrix with Uniform Local Binary Pattern Histogram, LBP histogram and Wavelet Transformed LBP histogram features are extracted from each image referring to total seven classes of protein. Synthetic Minority Over-sampling Technique (SMOTE) is used to remove class imbalance problem. K-fold cross-validation with three-fold was used to test the capability and efficiency of the dataset. The block diagram of the methodology used in this paper is given in Figure 2.

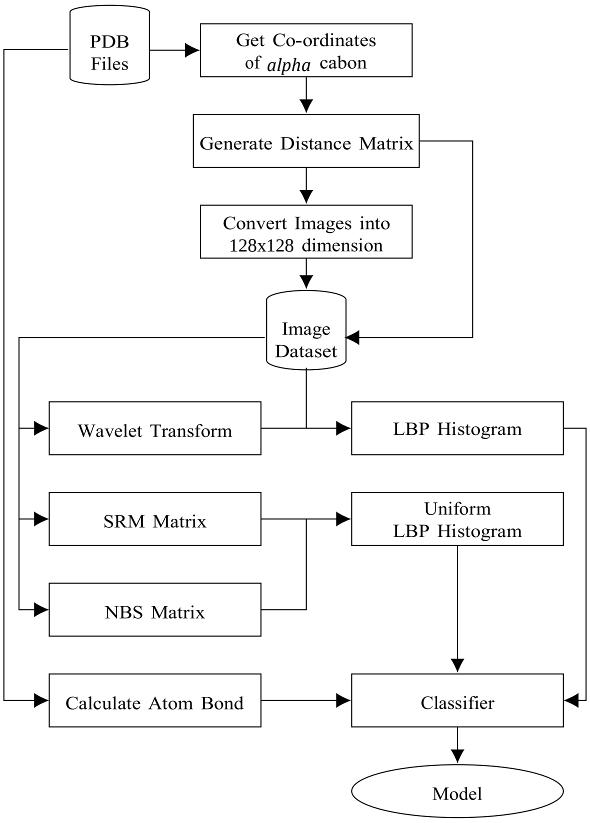


Figure 2. Block diagram of Protein Structural Class Prediction methodology

3.1.1 Dataset

We have used 40 percent ID filtered subset of PDB-style files for SCOPe domains version 2.03 [19] as our dataset. It contains a total of 12119 PDB files. Each PDB files contains SCOP(e) concise classification string (sccs) which respectively describes class, fold, superfamily, and family. In this literature, we are going to experiment only with the class of protein. In the dataset, there are total seven protein classes. The names of the protein classes can be found in Table 1.

Table 1. Protein Classes and its Corresponding Instances

Class Name	Total Instances
Small Proteins	640
All α Proteins	2195
α and β proteins(a/b)	3305
α and β proteins(a+b)	3006
Membrane and cell surface proteins and peptides	204
All β proteins	1485
Multi-domain proteins(α and β)	219

3.1.2 Image Generation

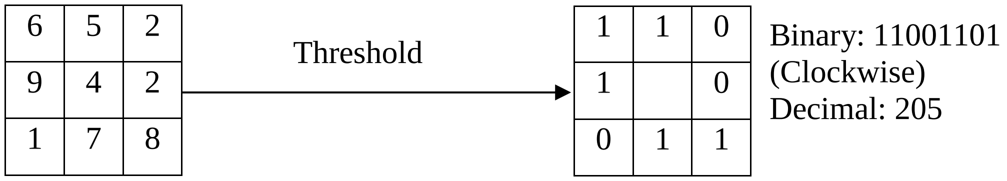
We have generated Images of Protein Structure according to the methodology described in CoMOGrad and PHOG [8]. The number of α carbons of protein can be found in the PDB file of the protein. The total number of α carbon atoms are calculated from the PDB file and the x, y and z coordinates of the α carbon stored. They are used to generate a distance matrix. The matrix is used as the image of the protein structure of that particular protein. The generated images are black and white in nature.

3.1.3 Scaling Images to Same Dimension

The dimension of protein images is based on the total number of α carbon they have. So, every individual protein image is different from the other. Therefore, the images should be scaled to the same dimension. CoMOGrad and PHOG have used Bi-cubic interpolation and wavelet transform to scale all the protein images into 128 x 128 dimension [8]. During the Bi-cubic interpolation step, most of the images were in 128x128 dimension so in the wavelet transform step they scaled all the images to that dimension. Thus, we have directly scaled the images to 128x128 dimension. We have used both real and scaled images to examine the results.

3.1.4 Feature Extraction

Our first four feature groups are types of histograms and the fifth feature group is about the prognosis of the atoms. The histograms were made from both scaled and unscaled images.



6	5	2
9	4	2
1	7	8

1	1	0
1		0
0	1	1

Figure 3. An example of basic LBP

1) Local Binary Pattern Histogram: The work of Ojala et al. [20] popularized LBP. Although it was first narrated in 1994 [21]. Local Binary Pattern computes the local representation of the texture of an image as a texture descriptor. Comparing each pixel with its neighboring pixels the local representation is created. The image is transformed into a grayscale image. In a 3x3 neighborhood, the center pixel value is calculated by comparing with its eight neighboring pixels. Each comparison gives a result of either 0 if the center pixel value is greater than the comparing neighbor pixel or 1 for the latter. A clockwise direction starting from the top-left one provides a binary number. The binary number is converted to a decimal number and the value is placed in the center pixel. LBP codes or Local Binary Patterns are the obtained binary numbers. An example of a basic Local Binary Pattern is given in Figure 3. After calculating the value for each pixel of the image, a histogram is calculated. A 3 x 3 neighborhood has 28 = 256 possible patterns, thus the values range from 0 to maximum 255 in each pixel of the image. The total number of bins of the histogram is thus 256. We would get 256 attributes from each image.

2) Wavelet transformed Local Binary Pattern Histogram (WtLBP-Hist): We have used Haar wavelet transform [22] for our wavelet transformation. It is based on lifting scheme. Wim Sweldens developed by Lifting scheme [23]. The image is converted into three two dimensional matrices for storing blue, green and red value of each pixel. The rows and columns of the three matrices and protein image are equal. Haar wavelet transformation is applied on the three matrices and the corresponding pixel value of the

three images are added to generate the wavelet image. LBP techniques are then used on wavelet image to get 256 attributes.

3) Separate Row Multiplication Matrix with Uniform Local Binary Pattern Histogram (SRM Matrix-ULBPHist): The image is split into 3x3 matrices. From each matrix, we get 3 rows with the dimension of 1x3. By multiplying each row with the same 3x3 matrix, we get three result matrices consisting of 1x3 dimension. Each cell is divided by 100. The results are then put in the 3x3 matrix in accordance with the row numbers. The color intensity of an image is between 0 to 255. So, if the value ofanycelloftheresultmatrixisgreaterthan255, then the value is replaced with 255. After applying this technique, the uniform local binary pattern is applied. From Figure 4, (a) presents a 3x3 section of matrix and the rows, (b) exhibits the result of multiplication, (c) shows the value after dividing by 100, (d) shows the replacement result of value greater than 255 and (e) shows a 3x3 matrix section after SRM-Matrix transformation. Another variation of the LBP is called uniform pattern [20]. Some binary patterns occur more generally in texture images. If the binary pattern comprises of at mosttwo0-1or1-0transitionswhenthebitpatternisheld circular then the pattern is called uniform. For instance, 01000000 has 2 transitions, 00000111 has 2 transitions which are uniform pattern on the other hand 01010100 has 6 transitions,11001001 has 4 transitions which are not uniform. A neighborhood with the dimension of 3x3 has 28 = 256 possible patterns with 58 of them being uniform. For estimating the histogram, every uniform pattern gets a separate bin while a single bin is allotted for all non-uniform patterns. Therefore, from a uniform binary pattern, we get the histogram of total bin size of 59.

4) Neighbor Block Subtraction Matrix with Uniform Local Binary Pattern Histogram (NBSMatrix-ULBPHist): Blocks are of the same dimension, 3x3. Two blocks of matrices are considered neighbors for this method if the center cells are neighboring. Because of this, the value of the last two columns of the first block and first two columns of the second block are same. The two blocks of matrices are subtracted and the result is set in the place of the first block. If any of the cells have any negative number, then 0 is placed instead of the negative value. The replacing of value is made because the histogram bin begins from zero. Uniform local binary pattern is then used to compute the histogram.

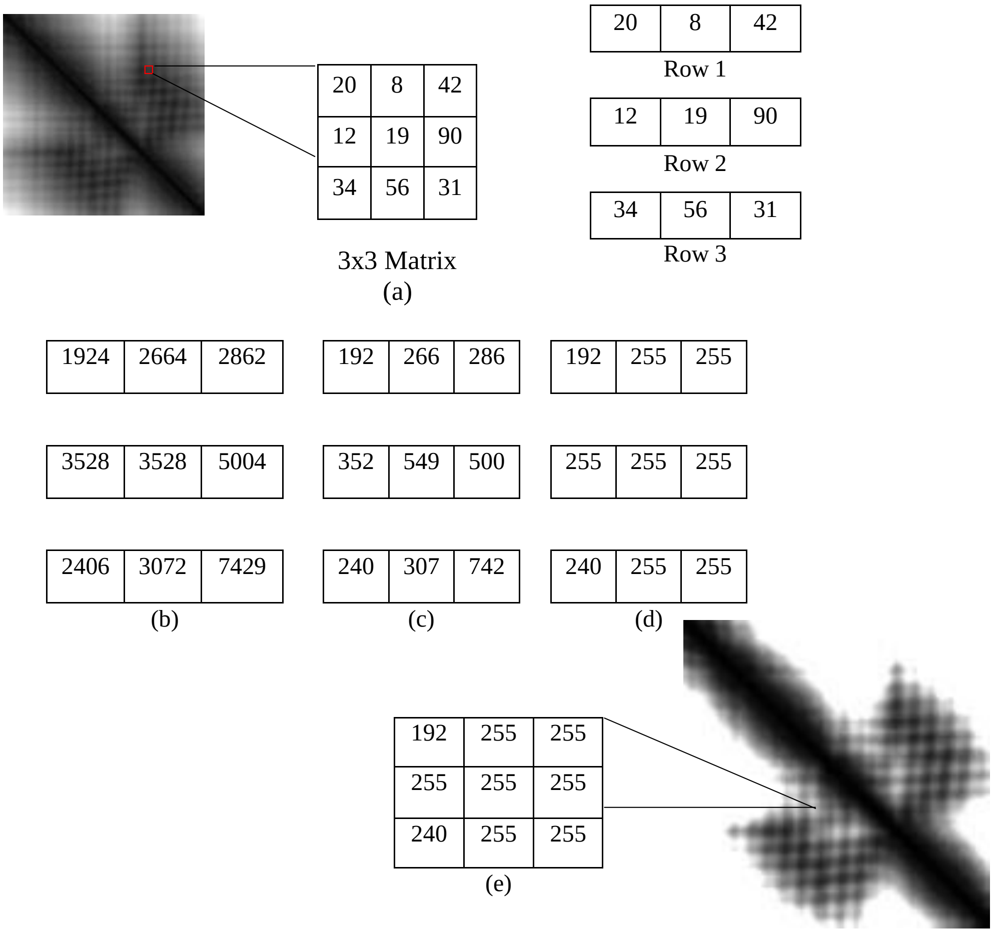


Figure 4. An example of Separate Row Multiplication Matrix with Uniform Local Binary Pattern Histogram

5) Atom Bond Features: First of all, we've identified unique atoms amidst all the protein PDB files. From each protein PDB file, we've counted occurrences of each atom. Then we've taken the percentage as features of each atom among all the atoms that each protein has. Then we've taken first 100 sequential atoms and used their atomic mass as the feature. Then we've counted the bond that each pair of atoms has in a particular protein using atomic distance based on a threshold value. Finally, we've taken the percentage as the feature of the bond of each unique pair of atoms among all the bonds that the protein has. Summary of all the feature groups used in this paper is given in Table 2.

Table 2. Feature Group For Protein Class Prediction

Identifier	Feature Group Name	Number of Features
A	LBP-Hist	256
B	WtLBP-Hist	256
C	SRM Matrix-ULBP-Hist	59
D	NBS Matrix-ULBP-Hist	59
E	Atom Bond	116

3.1.5 Re-evaluate Dataset

To perform benchmark analysis, we received the dataset generated from the CoMOGrad and PHOG literature [8]. As the SCOPe-sid is unique for every variant of the protein, we have created the dataset based on the proteins which are both on the 40 percent Id filtered subset and CoMOGrad and PHOG paper. After analyzing we have found that there are total 11052 instances in both of our feature groups and CoMOGrad and PHOG features. The seven classes and the total number of instances of each class are given in Table 1.

3.1.6 Removing Multiclass Imbalance Problem

From Table 1, it can be noted that the classes are imbalanced. To balance the classes, we have used Synthetic Minority Over-sampling Technique (SMOTE) [24].

In Weka, the percentage of SMOTE indicates that how many more instances would be generated. As the highest number of instances, a class has is 3305, we have oversampled our instances close to that number. If x is denoted by the highest number of instances among all the classes and y denoted by a class which we will SMOTE then the equation for the percentage calculation is shown in (1).

$$\frac{x-y}{y} * 100 \qquad (1)$$

We have used 5 nearest neighbors to generate the oversample dinstances. After applying SMOTE to all datasets, the total number of instances of each dataset close to 23132.

3.1.7 Classifiers

We have used five classifiers: K-Nearest Neighbor (KNN), Naive Bayesian Classifier, Support Vector Machines (SVM), Adaptive Boosting (AdaBoost) and Random Forest. A concise description of the classifiers is given in this section.

1) K-Nearest Neighbor (KNN): K-nearest neighbor algorithm (KNN) [25] is a similarity-based classification technique. It is a lazy classification technique. Distance metrics are used for each instance of the whole dataset for calculating the K nearest neighbors. The labels of the nearest neighbors decide the label of the test instances. It works poorly for high dimensional data. Euclidean distance, Hamming distance, Manhattan distance, Minkowski distance, Tanimoto distance and Jaccard distance are used for similarity measures.
2) Naive Bayesian Classifier: Naive Bayesian classifier [25] is based on probabilistic inference of samples observed where the decision variable and the features form a very naive structure of Bayesian Network. Naive Bayesian classifiers work best for image recognition and text mining.
3) Support Vector Machine (SVM): Support Vector Machine [25] works by creating and separating hyperplane for a given dataset by sampling different classes which are separated by maximum width.
4) Adaptive Boosting (AdaBoost): Adaptive Boosting classifier [25] is a meta-classifier which aims to make a strong classifier using a set of weak classifiers. The classifiers whose performance are marginally better than random classifiers are called weak classifiers.
5) Random Forest: Random Forest [25] is an ensemble classifier. A decision tree is created in each iteration with features taken randomly. It samples selected features using bootstrap aggregating.

3.1.8 Performance Evaluation

Separate independent test set or cross fold sampling method is used by researchers for performance evaluation. They are used to check the stability of the model. As k-fold cross-validation overcomes the problem of over-fitting it is preferred by researchers for performance approximation. We have used k-fold cross-validation technique. K-fold cross validation splits the data into k partitions and then use each partition as a test set with each iteration where the training data is the rest of the data. We have used accuracy as the performance metric in this paper. The percentage of correctly classified instances to the total number of instances is termed as accuracy.

3.2 Protein-Ligand Binding Prediction

Protein Ligand Binding prediction is a binary class classification problem. We've used Image Based Features for each Protein and Ligand dataset. Our methodology is a weak learner as it doesn't make any model. It is Based on threshold values.

3.2.1 Dataset

We've used Computer Vision and Pattern Discovery for Bioimages Group @ BII as our dataset. In our dataset, there are 3000 protein-ligand complexes that were determined experimentally with 3D structures available. Each protein (xxxx_pro_cg_.pdb) and its ligand (xxxx_lig_cg_.pdb) are of one-to-one correspondence, i.e. they can bind to each other and make Protein-Ligand complex. The dataset has 3000 pairs of protein and ligand where same name/ID of protein and ligand interacts/binds with each other.

3.2.2 Data pre-processing

We've used OpenCV [7] library to create images from PDB files. For protein, we've considered the coordinates of only the alpha-carbons to generate the distance matrix to create image. Because alpha-carbon can represent the structural information of protein quite well. But the given ligands were small in terms of atom number. So, while creating ligand images, we've considered all the atom's co-ordinates for generating distance matrix.

Among the PDB files, 33 ligands have only one atom, which will create 1x1 image having no significance for feature extraction. So, we had to compromise those 33 ligands as well as 33 corresponding proteins from training.

The given dataset has only positive instances (the pairs of protein and ligand where they bind with each other). But there were no negative instances (the pairs of protein and ligand where they do not bind with each other). The missing negative instances have created our dataset highly imbalanced. To overcome this imbalance, we've generated negative instances in two different ways.

1) Random Negative Undersampling: We have 2967 protein PDB and 2967 ligand PDB where 8803089 pairs are possible. Among these, 2967 pairs are given as positive instances and the rest 8800122 pairs are unknown/unseen instances. From the unseen pairs, we've taken 2967 pairs randomly as negative instances to make our dataset balanced.

2) Clustering-Based Undersampling: Using the positive instances (2967 pairs), we've created 10 clusters. Then we've searched for 2967 unseen pairs randomly as negative instances where they belong to those 10 clusters. We've made sure that each cluster has exactly same number of positive and negative instances to make the dataset balanced. Shown in Figure 5.

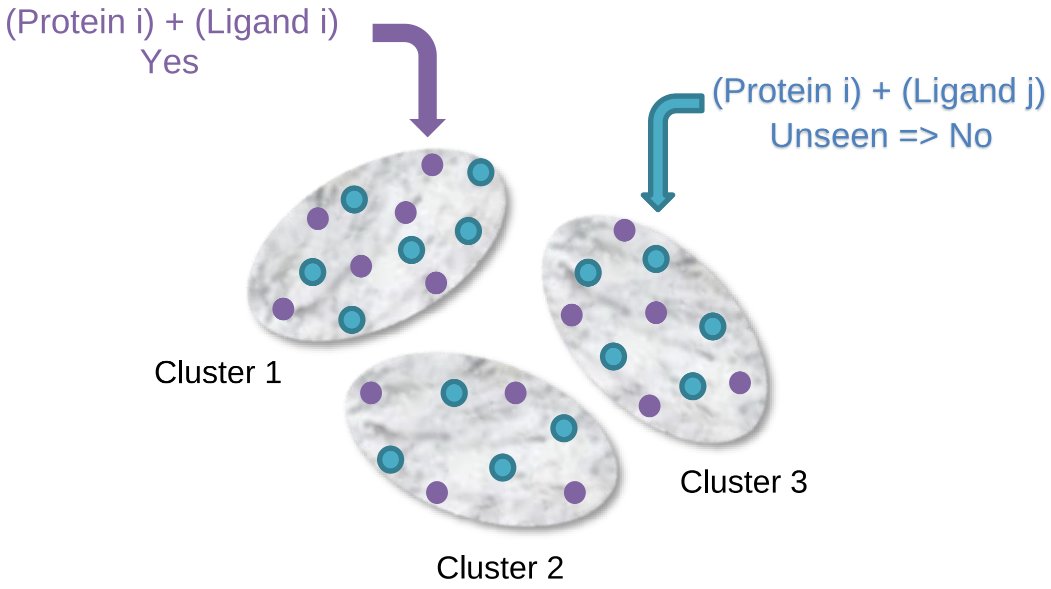


Figure 5. Clustering-Based Undersampling

3.2.3 Our Classifier

We’ve developed a similarity-based clustering method to predict the binding class. Distance is used to measure similarity. Our methodology is given in Figure 6.

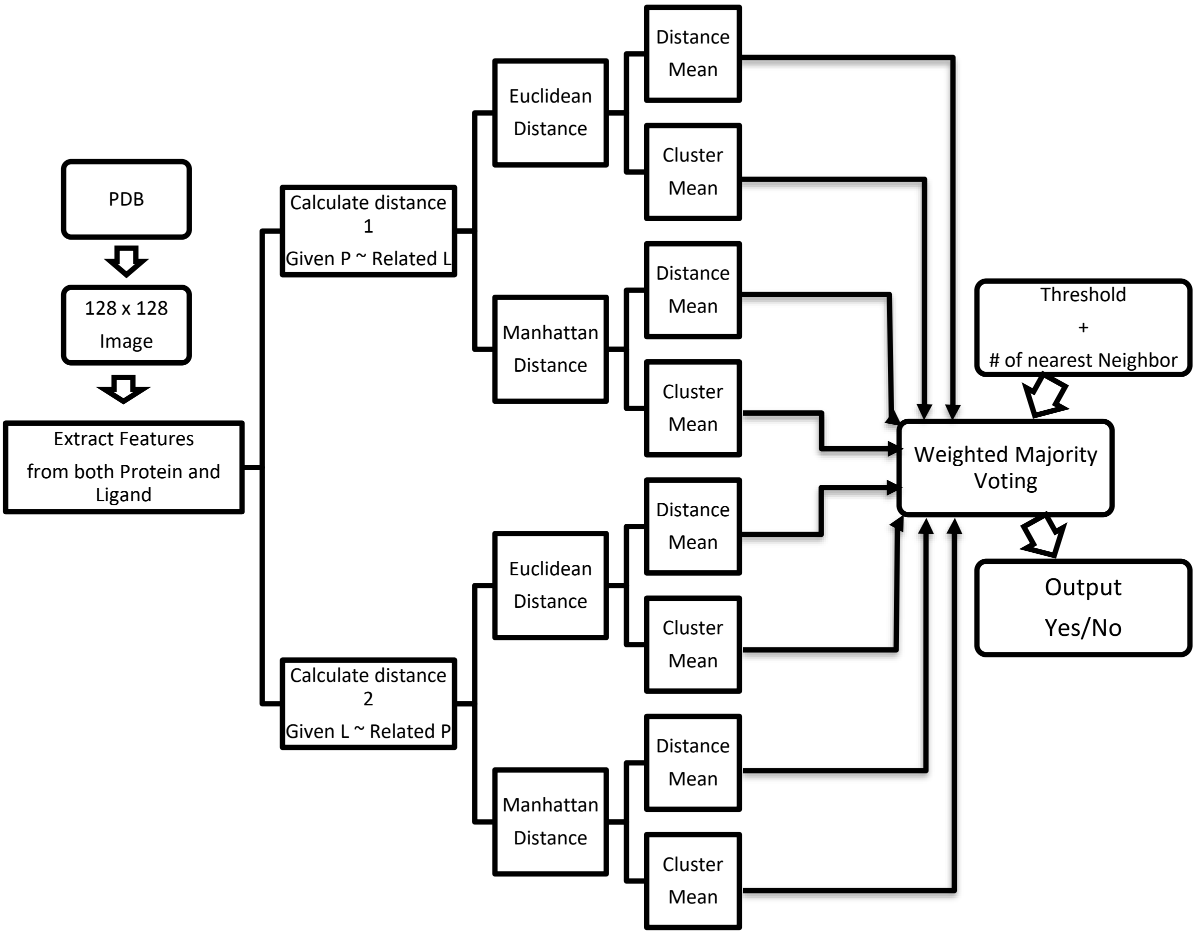


Figure 6. Block Diagram of Similarity Based Clustering for Protein-Ligand Binding Prediction

Pseudocode:

```
for all proteins & ligands
    generate images & extract features
end for
for each of the given pairs of protein-ligand
    NP = k-nearest protein(s) of the given protein
    RL = k related ligand(s)
    D1 = distance between given ligand & RL
    if D1 < Threshold1
        V1 = vote for positive bind
    else
        V1 = vote for negative bind
    end if
    NL = k-nearest ligand(s) of the given ligand
    RP = k related protein(s)
    D2 = distance between given protein & RP
    if D2 < Threshold2
        V2 = vote for positive bind
    else
        V2 = vote for negative bind
    end if
    MV = weighted majority voting between V1 & V2
end for
```

From the PDB dataset firstly we've generated images and converted to 128 x 128 images for each protein and ligand. From these images we've generated 2 different features.

1) *CoMOGrad and PHOG*: CoMOGrad stands for Co-occurrence Matrix of the Oriented Gradient of Distance Matrices and PHOG stands for Pyramid Histogram of Oriented Gradient [6]. This methodology also uses the α carbon distance matrix of protein. The dimension of all distance matrix is converted to 128 × 128. In CoMOGrad, the gradient angle and magnitude is computed from the distance matrix and the values are quantized. Quantization is a compressing technique which compresses a range of values to a single quantum value. In this methodology, the values are quantized to 16 bins which produce a co-occurrence matrix which is 16 × 16 matrix. The matrix is converted into a vector of size 256. Quadtree from the distance matrix is created with the desired level in PHOG. Gradient Oriented Histogram of each node is calculated with the preferred number of bins and bin size. In gradient oriented histogram an image is divided into small sub-images called cells and histogram of edge orientations are accumulated within the cell. The combined histogram entries are used as the feature vector describing the object. Total features which are the multiplication of total nodes and number of bins are

incorporated in the vector with the size of the total number of features. The vector is normalized by dividing it with the sum of its components.

2) *Hybrid Local Binary Pattern (LBP): Hybrid* Local Binary Pattern is a procedure of local binary pattern histogram, Wavelet transformed Local Binary Pattern Histogram, Separate Row Multiplication Matrix with Uniform Local Binary Pattern Histogram and Neighbor Block Subtraction Matrix with Uniform Local Binary Pattern Histogram for protein structural class prediction that we've generated earlier. These are on the distance matrix of α carbons of proteins which are used as an image for feature extraction.

Distance can only be calculated between proteins or between ligands. We've used K-nearest neighbor and Clustering method to calculate these distances.

i. *Related Ligand(s):* For a given Protein, find K-nearest proteins. The ligands those binds with the above nearest proteins, are the Related Ligands for the given protein.

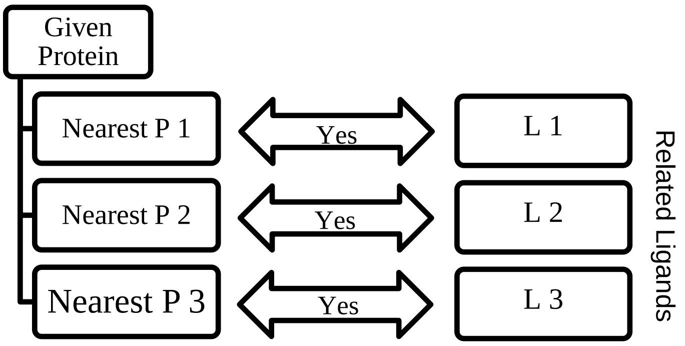


Figure 7. Relation between given Protein and Related Ligands

ii. *Related Protein(s):* For a given Ligand, find K-nearest ligands. The proteins those binds with the above nearest ligands, are the Related Proteins for the given ligand.

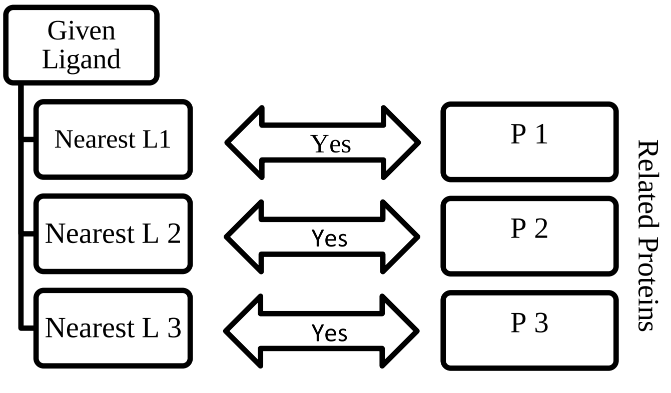


Figure 8. Relation between given Ligand and Related Proteins

iii. *Distance 1 (Given Protein ~ Related ligand): D*istance between related ligand and given ligand.

iv. *Distance 2 (Given Ligand ~ Related Protein): D*istance between related protein and given protein.

v. *Distance formula:* two types of distance can be measured. Euclidean (2) and Manhattan (3) distance.

$$D_{ij} = \sqrt{\sum_{k=1}^{n} \left(x_{ik} - x_{jk}\right)^2} \tag{2}$$

$$D_{ij} = \sum_{k=1}^{n} \left|x_{ik} - x_{j_k}\right| \tag{3}$$

vi. *Distance Mean:* Mean of distances between given instance and each of the related instances.

vii. *Cluster Mean Distance:* Distance between given instance and the cluster mean of the related instances.

viii. *Threshold:* Threshold is the boundary between similarity and dissimilarity in terms of distance. If distance is less than the threshold, then prediction in positive similarity, else the prediction is negative similarity. Threshold of each category of distances is the average of minimum and maximum distance based on the number of nearest neighbors.

For a given pair of Protein and Ligand, we want to predict if the will bind with each other or not. For measuring Distance 1, from the given protein, we searched for 5/3-nearest proteins and found the 5/3 Related Ligands accordingly. Then we've calculated the distance using above mentioned methods. Then we've taken the vote for the binding class by all categories of distances based their thresholds. Then finally, we've used weighted majority voting mechanism to predict the binding class.

Chapter 4

Results

4.1 Protein Structural Class Prediction

In this section, we are going to describe the experiments conducted in this study. Some of the experiments were carried out in a Dell Inspiron 15 Laptop Computer of 3000 series with 4 GB RAM and 240 GB SSD hard drive, some of them were carried out in Intel Core i5 4590 processor personal computer with 12 GB Ram and 2048MB ATI AMD Radeon R7 200 Series Graphics Card and others were experimented in a Computing Machine provided by CITS, United International University which was equipped with 8 core processors each having a Dell R 730 Intel Xeon Processor (E5-2630 V3) with 2.4 GHz speed and 18.5 GB memory. Java language is used to implement all the programs using the Eclipse IDE and Java 8 standard edition. Features were generated using The OpenCV software library [26]. Weka tool [27] was used to implement the classification algorithms used in this paper.

4.1.1 Parameters used for the classifiers

A different set of parameters were used for each classifier. A linear searching was used with no distance weighting for KNN. In case of the Naive Bayesian Classifier, SVM, a polynomial kernel was used with c = 1.0 and $\epsilon = 1.0w{-}2$. Data was normalized before supplying to the classifier. J48 decision tree classifier was used in AdaBoost classifier as the weak base classifier. Classifier number of iterations was set to 100 for Random Forest.

4.1.2 Analysis of Features

In this section, we are we are going to present the analysis of our features. Results in terms of average accuracy in 3-fold cross-validation of protein images are given in Table 3. The highest percentage of correctly classified instances achieved for each of the classifiers are indicated by the boldly faced values of the table.

After running the experiments for our five feature groups ABCDE classifies the highest percentage of correct instances in Random Forest, Adaboost and SVM among all other feature groups. Feature group CD provides the highest accuracy in KNN and Naive Bayesian. As the whole combination of all feature groups accuracy gives the highest percentage than any other feature group, thus we conclude that the best performing feature group combination is ABCDE and the best classifier is Random Forest classifier.

Table 3. Accuracy of different classifiers of protein images

Image Type	Feature Type	Classifiers				
		KNN	Naïve Bayesian	SVM	AdaBoost	Random Forest
Non Scaled	A	68.69	33.15	65.62	83.37	87.50
Non Scaled	B	77.28	32.17	67.24	83.52	86.88
Scaled	A	74.06	53.06	78.58	83.30	85.22
Scaled	B	78.69	49.79	79.80	84.68	86.68
Scaled	C	84.10	51.02	71.02	83.20	85.11
Scaled	D	81.58	56.40	72.25	81.89	83.99
	E	66.96	21.79	44.49	62.26	69.92
Scaled + Non Scaled	AB	78.62	41.87	83.08	83.08	86.08
Scaled	CD	**84.65**	**57.55**	78.10	84.24	86.28
Scaled + Non Scaled	ACD	84.51	41.96	81.99	86.97	88.66
Scaled	BCD	81.87	55.02	84.29	85.64	87.37
Scaled + Non Sclaed	ABCD	84.38	36.90	83.53	86.48	88.77
Scaled + Non Sclaed	ABCDE	76.47	36.86	**85.78**	**87.57**	**89.03**

4.1.3 Comparison with other methods

In this section, we compare the performance of our proposed method with CoMOGrad and PHOG[8] along with our previous published literature Wavelet and Pyramid Histogram Features for Image Based Leaf Detection[11]. For comparison with our methodology in this literature, we applied CoMOGrad and Phog techniques and Wavelet and Pyramid Histogram techniques in our dataset of 11052 instances and later applied SMOTE for reducing class imbalance problem. We conducted experiments with different classifiers using the same parameters as we did for feature analysis with the feature groups. The results are given in Table 4. From Table 4 it can be comprehended that our feature group ABCDE outperforms CoMOGrad and PHOG in Random Forest and in Adaboost. CoMOGrad and PHOG surpassed our feature groups in KNN, Naive Bayesian and SVM. It can be noted that the combination of our feature groups are three-fourths of

CoMOGrad and PHOG. It also can be discerned that the accuracy percentage in Random Forest is higher than all the classifier results. Thus, our novel features can classify more instances than CoMOGrad and PHOG. We have also noticed that our feature groups outperform the features of our previous literature [11] on all classifiers.

Table 4. Comparison of the method proposed features in this paper with [8] and [11]

Feature Type	Classifiers				
	KNN	Naive Baysian	SVM	Ada Boost	Random Forest
Karim et al.[8]	87.41	59.50	87.67	84.19	85.49
Ahmed et al.[11]	69.36	36.22	67.30	79.92	84.58
this paper	84.65	57.55	85.78	87.57	89.03

4.1.4 Discussion

We have revealed the precedence of our methodology over CoMOGrad and PHOG [8] and Wavelet and Pyramid Histogram Features for Image Based Leaf Detection [11]. The same feature groups were used for leaf detection [11] with the dataset consisting of RGB images of leaves. Unlike only gray histogram used on this paper, blue, green and red histograms were used to generate features in each feature group and the accuracy result of each classifier was high. The distance matrix of α carbons or the protein images were black and white, thus only gray histogram was used as a feature.

We also used Scale-invariant feature transform (SIFT) [28] methodologies in our experiments. Each descriptor has a 128-dimensional feature vector. The number of the descriptors of SIFT from every image is not specific so we cannot use traditional machine learning techniques. Hence to apply traditional machine learning procedure and specify the feature vector, we split the image into 16 slices and took one descriptor from each of the slice images. Therefore, we got 2048 number of attributes(8x16) from each image. We tested the dataset with the same classifiers mentioned in this paper. The results didn’t turn up to be better or close to our proposed methodology in this literature.

4.2 Protein-Ligand Binding Prediction

This section is the description of our experiments performed in this study. Some of the experiments were carried out in a personal desktop computer having Intel Core i3 and 4 GB RAM and others were experimented in a Computing Machine provided by CITS, United International University which was equipped with 8 core processors each having a Dell R 730 Intel Xeon Processor (E5-2630 V3) with 2.4 GHz speed and 18.5 GB memory. Java language was used for data preprocessing including feature generation using OpenCV software library[7], negative data generation and data merging using Eclipse IDE with Java 8 standard edition. Python language was used to implement our algorithm using the Spyder IDE. Weka tool was used to run the traditional classification algorithms for the comparison with our algorithm. We've used Leave-One-Out validation method to get the accuracy of our model.

4.2.1 Dependencies/Hyperparameters

A. Number of nearest neighbors: Our algorithm's prediction accuracy is highly dependent on the number of nearest neighbors for finding both Related Protein(s) and Related Ligand(s). We've used 5 nearest neighbors in this experiment.

B. Threshold: This is the threshold of distance for determining whether two proteins or two ligands are similar or not. For a higher value of threshold, there is a higher possibility for our algorithm to predict positive binding class for the majority of the Protein-Ligand pairs. And the lower the threshold is, the higher is the possibility of negative binding class prediction. We've taken the average of distances among 5 nearest neighbors as our threshold for each category of the distances.

4.2.2 Classifiers

We've used traditional machine learning classifiers on the image-based feature dataset to compare with our algorithm. Each of them was executed using 10-fold validation. By merging both protein and ligand features and adding binding class at the end, we've created instances for each of the Protein-Ligand pairs. Used classifiers are described below.

1) Adaptive Boosting (AdaBoost): Adaptive Boosting [8] classifier is a meta-classifier which aims for more accurate classification using a group of weak classifiers. The weak classifiers are called base classifiers. The accuracy of these weak classifiers gradually increases as it acquires knowledge from the previous classifier/iteration. Overall classification is generated by weighted voting between the base classifiers.

Here, we've used J48 as base classifier which is actually the implementation of C4.5 algorithm.

2) K-Nearest Neighbor (KNN): KNN [8] is a weak classifier which is based on similarity measurement. Class label of test instance is the majority class label of the closest K-neighbors of the training data. Euclidian distance, Manhattan distance, Hamming distance etc. are used for similarity measurement.
3) Random Forest: Random Forest [8] is an ensemble classifier which creates decision tree by taking random features. It samples selected features using bootstrap aggregation.
4) Support Vector Machine (SVM): SVM [8] creates a hyperplane between different class samples by maximizing the separation width. It classifies test instances using that hyperplane/separation line.
5) Naïve Bayesian: Naïve Bayesian [8] algorithm is actually a probabilistic classifier which uses Bayes' theorem with strong (naive) independence assumption between the features.

4.2.3 Comparison with other Classifiers

a) With negative data: As we've generated negative data for solving the imbalance problem. But we're not sure if those negative data are actually negative or not. This was our assumption that all unseen pairs of Proteins and Ligands are of negative class. So, these are actually noisy data and will result in low accuracy. But we've used this noisy data as our advantage to get the threshold that determines the similarity. In this case, Distance1 (Given L ~ Related L) -> Manhattan Distance -> Distance mean works best. In spite of low performance with negative data, our algorithm works better than other existing widely used algorithms shown in Table 5 and Chart I.
b) Without Negative Data (Sensitivity): Sensitivity is the true positive rate regarding the positive instances. As we had to generate the negative data artificially, sensitivity is the actual scale of performance measuring where positive data were the actual data. In this case, Weighted voting of both Distance1 (Given L ~ Related L) and Distance2 (Given P ~ Related P) based on Manhattan Distance -> Cluster mean works best. Using the thresholds gained using the negative data, sensitivity of our algorithm is very good comparing to other existing algorithms shown in Table 6 and Chart II.

Table 5. Accuracy Comparison Table

Features	AdaBoost (J48)	KNN (5)	Random Forest	SVM	Naïve Bayesian	Our Method (5)	Our Method (3)
LBP (random)	40.70%	36.50%	31.85%	39.87%	43.98%	56.07%	**56.99%**
LBP (cluster)	43.76%	45.42%	40.77%	41.14%	49.06%	55.65%	**58.11%**
CoMOGrad & PHOG (random)	49.17%	47.54%	14.56%	34.26%	47.61%	**59.03%**	57.33%

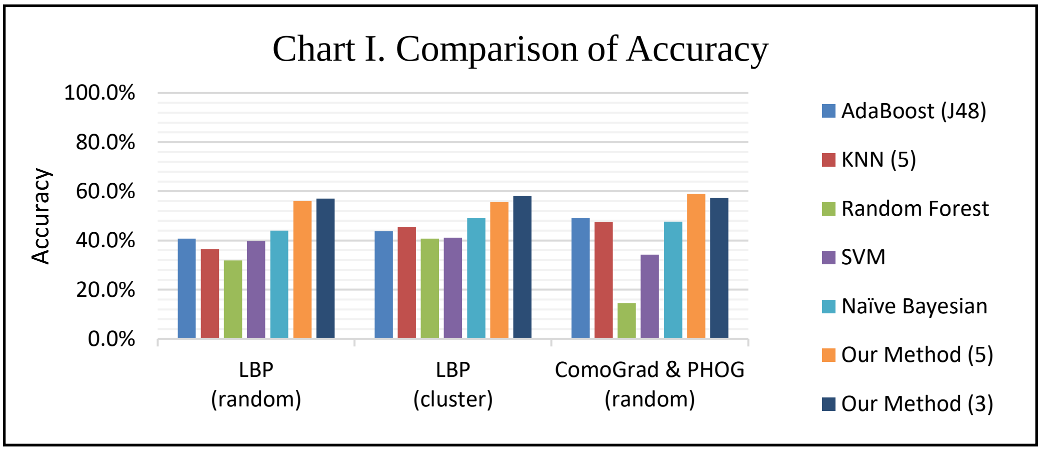


Chart I. Comparison of Accuracy

Table 6. Sensitivity Comparison Table

Features	AdaBoost (J48)	KNN (5)	Random Forest	SVM	Naïve Bayesian	Our Method (5)	Our Method (3)
LBP (random)	38.50%	36.80%	33.60%	43.80%	54.70%	84.40%	**84.50%**
LBP (cluster)	43.76%	38.40%	41.50%	37.40%	43.60%	83.69%	**84.37%**
CoMOGrad & PHOG (random)	**95.20%**	47.60%	16.10%	29.70%	11.30%	79.86%	78.31%

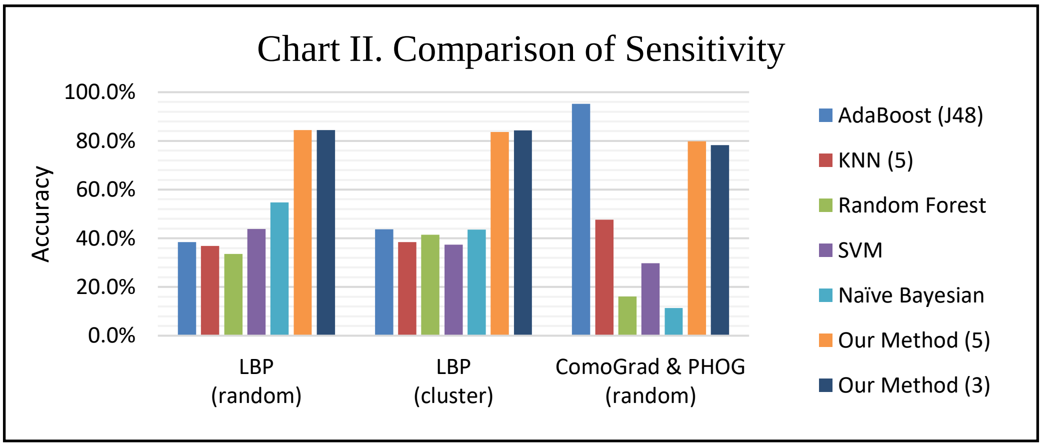


Chart II. Comparison of Sensitivity

We have generated three different datasets based on three different features. Hybrid LBP gives 630 long feature vectors from image. So, for one protein-ligand pair we've got 1260 (630+630) attributes and one Binding Class value as one instance. The above mentioned two types of negative data (random and Clustering-Based Undersampling) were generated using Hybrid LBP for balancing the data. CoMOGrad and PHOG gives 1021 or 1020 long feature vectors from protein image, but for ligand images, it gives 1020 long feature vectors. We assumed "0" as the last feature in protein where features were 1020 long, to make it 1021 long feature. So, for one protein-ligand pair we've got 2041 (1021+1020) attributes and one Binding Class value as one instance. Random negative undersampling was used in CoMOGrad and PHOG but Clustering-Based Undersampling was not possible as some clusters couldn't get any unseen pairs of protein and ligand. Our method was used based on 5 and 3 nearest neighbors and shown on the above tables and charts.

We can see that; AdaBoost works better than our algorithm in terms of sensitivity in ComoGrad and PHOG dataset. Because, Ligand data were so small in terms of number of atoms that ComoGrad and PHOG gave zeros for most of the ligands. But our algorithm's overall performance is better than other machine learning algorithms in the three different feature datasets

Chapter 5

Conclusion

5.1 Protein Structural Class Prediction

In this thesis, we showed how accurately we can detect protein classes using the combination of our feature group ABCDE of protein images. As the advancement of deep learning, neural network, and many other deep learning techniques are being used to classify images, many remarkably interesting applications can be made. For our future advancement, we wish to introduce new features to improve accuracy, use new tools and explore other fields of computer vision such as human emotion detection.

5.2 Protein-Ligand Binding Prediction

We are proposing a simple similarity-based clustering method to predict Protein-Ligand Binding without using deep-learning, neural-network. This simple distance-based algorithm is quite effective compared to complex machine learning algorithms.

Our main limitation was the missing negative data. If we had the actual negative data, we could've determined the perfect thresholds for each category of distances, and that would give us more accurate prediction. Another problem was dimensions of small Ligands as we're using image-based features.

In future, we will try to extract some unique features from the Ligand dataset so that the dimensionally problem doesn't affect our Protein-Ligand binding prediction.

References

[1] C. Chothia and A. M. Lesk, “The relation between the divergence of sequence and structure in proteins.” The EMBO journal, vol. 5, no. 4, pp. 823–826, 1986.

[2] G. P. Brady and P. F. Stouten, “Fast prediction and visualization of protein binding pockets with pass,” Journal of computer aided molecular design, vol. 14, no. 4, pp. 383–401, 2000.

[3] L. Holm and C. Sander, “Protein structure comparison by alignment of distance matrices,” Journal of molecular biology, vol. 233, no. 1, pp. 123–138, 1993.

[4] W. R. TAYLOR, “Protein structure comparison using iterated double dynamic programming,” Protein Science, vol. 8, no. 3, p. 654–665, 1999.

[5] S. Srivastava, S. B. Lal, D. Mishra, U. Angadi, K. Chaturvedi, S. N. Rai, and A. Rai, “An efficient algorithm for protein structure comparison using elastic shape analysis,” Algorithms for Molecular Biology, vol. 11, no. 1, p. 27, 2016.

[6] L. Holm and C. Sander, “Dali/fssp classification of three-dimensional protein folds,” Nucleic acids research, vol.25, no.1, pp. 231–234, 1997.

[7] A. P. Singh and D. L. Brutlag, “Hierarchical protein structure superposition using both secondary structure and atomic representations.” in Ismb, vol. 5, 1997, pp. 284–293.

[8] R. Karim, M. M. A. Aziz, S. Shatabda, M. S. Rahman, M. A. K. Mia, F. Zaman, and S. Rakin, “Comograd and phog: From computer vision to fast and accurate protein tertiary structure retrieval,” Scientific Reports, vol. 5, pp. 13275 EP –, Aug 2015, article. [Online]. Available: http://dx.doi.org/10.1038/srep13275

[9] C.-R. Shyu, P.-H. Chi, G. Scott, and D. Xu, “Protein dBs: a real time retrieval system for protein structure comparison,” Nucleic Acids Research, vol. 32, no. suppl_2, pp. W572–W575, 2004.

[10] P.-H. Chi, G. Scott, and C.-R. Shyu, “A fast protein structure retrieval system using image-based distance matrices and multidimensional index” International Journal of Software Engineering and Knowledge Engineering, vol. 15, no. 03, pp. 527–545, 2005.

[11] A. A. N. Ahmed, H. M. F. Haque, A. Rahman, M. S. Ashraf, and S. Shatabda, “Wavelet and pyramid histogram features for image-based leaf detection,” in Emerging Technologies in Data Mining and Information Security, A. Abraham, P. Dutta, J. K. Mandal, A. Bhattacharya, and S. Dutta, Eds. Singapore: Springer Singapore, 2019, pp. 269–278.

[12] I. N. Shindyalov and P. E. Bourne, “Protein structure alignment by incremental combinatorial extension (ce) of the optimal path.” Protein engineering, vol. 11, no. 9, pp. 739–747, 1998.

[13] C. A. Orengo and W. R. Taylor, “[36] ssap: sequential structure alignment program for protein structure comparison,” in Methods in enzymology. Elsevier, 1996, vol. 266, pp. 617–635.

[14] Y. Ye and A. Godzik, “Flexible structure alignment by chaining aligned fragment pairs allowing twists,” Bioinformatics, vol. 19, no. suppl_2, pp. ii246–ii255, 2003.

[15] M. Shatsky, R. Nussinov, and H. J. Wolfson, “Flexible protein alignment and hinge detection,” Proteins: Structure, Function, and Bioinformatics, vol. 48, no. 2, pp. 242–256, 2002.

[16] Y. Zhang and J. Skolnick, “Tm-align: a protein structure alignment algorithm based on the tm-score,” Nucleic acids research, vol. 33, no. 7, pp. 2302–2309, 2005.

[17] L. Zhang, J. Bailey, A. S. Konagurthu and K. Ramamohanarao, “A fast indexing approach for protein structure comparison,” BMC bioinformatics, vol. 11, no. 1, p. S46, 2010.

[18] Y. Yang, J. Zhan, H. Zhao and Y. Zhou, “A new size independent score for pairwise protein structure alignment and its application to structure classification and nucleic-acid binding prediction,” Proteins: Structure, Function, and Bioinformatics, vol. 80, no. 8, pp. 2080–2088, 2012.

[19] 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.

[20] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Transactions on pattern analysis and machine intelligence, vol. 24, no. 7, pp. 971–987, 2002.

[21] T. Ojala, M. Pietikainen, and D. Harwood, “Performance evaluation of texture measures with classification based on kullback discrimination of distributions,” in Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on, vol. 1. IEEE, 1994, pp. 582–585.

[22] R. S. Stanković and B. J. Falkowski, “The haar wavelet transform: its status and achievements,” Computers & Electrical Engineering, vol. 29, no. 1, pp. 25–44, 2003.

[23] W. Sweldens, “The lifting scheme: A construction of second-generation wavelets,” SIAM journal on mathematical analysis, vol. 29, no. 2, pp. 511–546, 1998.

[24] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” Journal of artificial intelligence research, vol. 16, pp. 321–357, 2002.

[25] M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of machine learning. MIT press, 2012.

[26] G. Bradski, “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools, 2000.

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

[28] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004.

[29] Chaires, J.B. Calorimetry and thermodynamics in drug design. Annu. Rev. Biophys. 2008, 37, 135–151

[30] Patching, S.G. Surface plasmon resonance spectroscopy for characterisation of membrane protein-ligand interactions and its potential for drug discovery. Biochim. Biophys. Acta 2014, 1838, 43–55

[31] Rossi, A.; Taylor, C. Analysis of protein-ligand interactions by fluorescence polarization. Nat. Protoc. 2011, 6, 365–387

[32] Sousa,S.F.;Ribeiro,A.J.;Coimbra,J.T.;Neves,R.P.;Martins,S.A.;Moorthy,N.S.;Fernandes,P.A.;Ramos,M.J. Protein-ligand docking in the new millennium-a retrospective of 10 years in the field. Curr. Med. Chem. 2013, 20, 2296–2314

[33] Steinbrecher, T.; Labahn, A. Towards accurate free energy calculations in ligand protein-binding studies. Curr. Med. Chem. 2010, 17, 767–785