2012

A model to predict and analyze protein-protein interaction types using electrostatic energies

Gokul Vasudev

A MODEL TO PREDICT AND ANALYZE PROTEIN-PROTEIN INTERACTION TYPES USING ELECTROSTATIC ENERGIES

by
Gokul Vasudev

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2012

A MODEL TO PREDICT AND ANALYZE PROTEIN-PROTEIN INTERACTION TYPES USING ELECTROSTATIC ENERGIES

by

Gokul Vasudev

APPROVED BY:

Dr. Siyaram Pandey, External Reader
Department of Chemistry and Biochemistry

Dr. Alioune Ngom, Internal Reader
School of Computer Science

Dr. Luis Rueda, Advisor
School of Computer Science

Dr. Xiaobu Yuan, Chair of Defense
School of Computer Science

03 August, 2012

Declaration of Co-Authorship

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

Prediction and analysis of types of protein-protein interactions (PPI) is an important problem in molecular biology because of its key role in many biological processes in living cells. In this thesis, I propose a model called PPIEE (Protein-protein interaction using electrostatic energies) to predict and analyze protein interaction types using electrostatic energies as properties to distinguish between these types of interactions. This prediction approach uses electrostatic energies for pairs of atoms and amino acids present in interfaces where the interaction occurs. Using this approach, the results on well-known datasets confirms that electrostatic energy is an important property to predict obligate and non-obligate protein interaction types. The classifiers used are support vector machines and linear dimensionality reduction. Since electrostatic interactions are long ranged, some other experiments are performed by changing the threshold values, which are the distances calculated between atom pairs of interacting chains, ranging from 7Å to 13Å. This information will be helpful for researchers to understand how different physiochemical properties contribute to understanding about stability of protein complexes and their function.

Dedication

I would like to dedicate this thesis to my parents and Sai Babaji. My father always wanted to see me reaching the highest levels of success. I am very happy today that I am able to him feel proud of myself. Also thanks to my maa for all her prayers. It is because of their good wishes, I am able to do this thesis work. I would also like to mention my elder brother Mudit for his unconditional support.

Acknowledgements

I would like to take this opportunity to express my sincere gratitude to Dr. Luis Rueda, my supervisor, for his steady encouragement, patient guidance and enlightening discussions throughout my graduate studies. Without his help, the work presented here would have not been possible. Dr. Luis Rueda will always remain like a father figure to me.

I also wish to express my appreciation to Dr. Alioune Ngom, School of Computer Science and Dr. Siyaram Pandey, Department of Chemistry and Biochemistry for being in the committee and spending their valuable time. In addition, I would like to thanks Dr. Nathan Baker and his team for the tools, PDB2PQR and APBS, that helped me enormously in this thesis. Finally, in reviewing this thesis, I would like to thank Manish Kumer Pandit, Sonia Bhatti, Navid Shakibapour, Manoj Gajjarapu, Mina Maleki and all my friends for their consistent moral support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Protein-protein Interaction

Molecular biology is the study of biology at a molecular level. It overlaps biology and chemistry and studies vital processes and chemical substances in living organisms. This branch mainly deals with different types of interactions among various molecules such as different types of DNA, RNA and protein complexes. Understanding these interactions are also included in this branch of biochemistry. Molecular biology revealed the original convergence of geneticists, physicists and structural biochemists on a common problem; the structure and function of biological complexes. It focus heavily on function, role and structure of biomolecules. The history of molecular biology provides the importance of the discovery of macromolecular mechanisms [20]. Recently, a lot of work has been done at the interface of molecular biology and computer science in bioinformatics and computational biology.

Bioinformatics can be seen as an application of statistics, maths and computer science. It is actually, the application of computer science techniques to molecular biology [47], an

indispensable field for modern biology. Commonly used tools and technologies in this field are, Java, Python, Perl, Cuda, Octave, Matlab and Microsoft Excel among others. The main purpose of bioinformatics is to have better and lucid understanding of biology. It involves computationally intensive techniques such as pattern recognition, data mining, visualization and machine learning algorithms. Major research efforts are in the fields of structure prediction, drug design, drug discovery, protein structure alignment, gene expression and protein-protein interactions.

Proteins control all biological systems in a cell including nutrient uptake, gene expression, cell growth, proliferation, inter cellular communication, among others. The main reason behind their interaction is to perform some biological function. Many proteins perform their function independently. Prediction of protein-protein interaction (PPI) and analyzing relevant properties for prediction have been studied from various perspectives. Proteins bind to each other through a combination of hydrophobic bonding, van der Waals forces and salt bridges at specific binding domains on each protein. The strength of the binding is dependent on the size of binding domains. These domains can be large surfaces, small binding clefts, a few peptides or hundreds of amino acids.

Prediction of protein interaction has gained much interest in recent years with over 20 different proposed methods [37]. To characterize the properties of protein-protein interaction types can be done by studying their structural information. Thus, structure-based prediction methods including computational approaches, homology modelling, threading-based methods and protein-protein docking are more accurate than those which do not employ structural data [37]. These studies have been carried out mostly by relying on biological knowledge about the atoms or molecules, which normally are selected manually by observing groups of complexes or on prediction results.

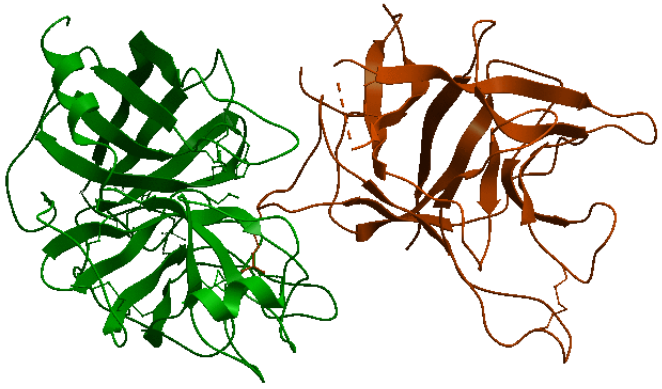


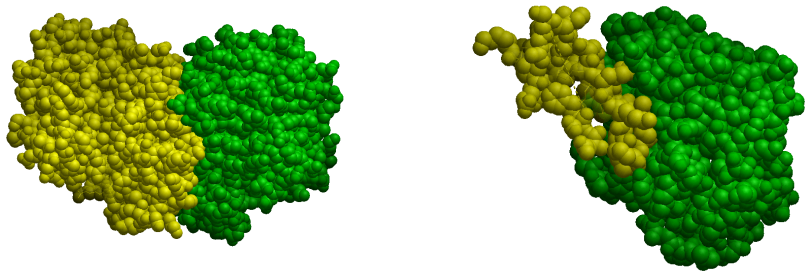
Figure 1.1: A non-obligate complex PDB ID 1avw with its interacting chains A and B, shown in different colors.

Figure 1.1 shows a non-obligate complex (PDB ID 1avw) an its two interacting chains with different colors. This figure was generated with the help of ICM browser [48]. It reflects how two chains of the protein complex interact with each other. The part of interest is the interacting atoms. The complex is taken from one of datasets used in the proposed model which is referred to as the Zhu dataset [50].

An important aspect in studying prediction of PPI is to predict different types of complexes, including similarities between subunits (homo/hetero-oligomers), number of subunits involved in the interaction (dimers, trimers, etc.), duration of the interaction (transient vs. permanent) [31], stability of the interaction (non-obligate vs. obligate) [50], among others; but I focus on the problem of prediction protein interaction types (obligate and non-obligate).

Obligate interactions are usually considered as permanent, while non-obligate interactions can be either permanent or transient [33]. Non-obligate and transient interactions are more difficult to study and understand due to their instability and short life, while obligate

and permanent interactions last for a longer period of time, and hence are more stable [22]. The study of [2] suggested that mobility differences of amino acids are more significant for obligate and large interface complexes than for transient and medium-sized ones.



(a) Obligate complex PDB ID 1luc (b) Non-obligate complex PDB ID 1tab

Figure 1.2: Differentiating between Obligate and Non-obligate protein complexes on the basis of the interface area. Part (a) of the figure shows an obligate complex while the other part (b) depicts a non-obligate complex.

Figure 1.2 depicts an obligate complex (PDB ID 1luc) and a non-obligate complex (PDB ID 1tab). It shows how obligate and non-obligate complexes differ from each other. Observing part (a), the two chains of the complex are shown in different colors, i.e, yellow and green. Since it is an obligate complex, it has larger interface area as compared to the other complex. On the other hand, it is a non-obligate complex (PDB ID 1tab) with two chains interacting with each other but with comparatively smaller interface area. Thus, on the basis of the interface area one would be able to differentiate between the two types of complexes. However, this is not the general case, since other properties may empower the prediction, as shown in this thesis.

Non-covalent interactions are very common between macromolecules (including proteins). There are three type of non-covalent interactions [23]:

i) Electrostatic interactions - they occur between electrically charged atoms having both positive and negative interactions.
ii) Vander waal interactions - they occur between any pair of charged atoms that are close to each other.
iii) Non-polar interactions - these are attractive interactions occurring between atoms that do not have charges.

Two atoms that are either partially or fully electrostatically charged will interact with each other. Each interacting atom generates an electric field that surrounds the atom. Also, there are a few types of electrostatic interactions which are dominant in proteins :
i) Ionic interactions: they occur between fully charged atoms.
ii) Hydrogen bonds: they occur between atomic dipoles. They involve atoms with partial charge, one of which is a hydrogen atom.

To predict the interaction types, every prediction method needs observed properties of the known class samples called $features$. Features are generally nominal or numeric, and the process of calculating the features for each sample from the input dataset is called *feature generation*. Physiochemical properties of proteins are very powerful for prediction and have been extensively reported in literature. Interacting regions can be characterized by diverse sets of physiochemical properties [37], topological properties [10] and conserved residues [27]. In addition, other properties have been used for PPI prediction such as analysis of solvent accessibility [50], geometry [25], hydrophobicity [44], sequence-based features and desolvation energy [4],[44]. Based on interface properties such as interface area and ratio of area [50], Zhu *et al.* predicted biological and crystal packing interactions using support vector machine (SVM). Using solvent accessible surface area and other interface

properties, prediction of types (obligate and non-obligate) was reported using SVM and linear dimensionality reduction (LDR) [43]. A very recent work presented by Rueda *et al.* shows the use of desolvation energies to solve the same prediction problem using the above mentioned classifiers [30].

To reduce the size of the generated features from the input, feature extraction methods have been used. Feature extraction is a popular pattern recognition method [18]. It is a special form of dimensionality reduction. When the length of the feature vector is very large, there might be a need to apply feature extraction methods to find the lower dimensional representation of that original feature vector. The transformation of high dimensional data to lower dimensional data is also called feature extraction [18]. There are many feature extraction methods such as principal component analysis (PCA) [21], non-linear dimension reduction, independent component analysis and linear dimensionality reduction (LDR). For classification purpose, I use LDR methods [41] coupled with Bayesian classifiers (details discussed in Chapter 3), which are the following:

- Fisher's discriminant analysis (FDA) [12, 15].
- Heteroscedastic discriminant analysis (HDA) [28].
- Chernoff discriminant analysis (CDA) [41].

Lower dimensional data obtained by these three LDR methods are then passed through quadratic Bayesian (QB) and linear Bayesian (LB) classifiers [12] for final prediction. For each classifier, prediction accuracy and time taken for that prediction are very important. It actually depends on the dataset used as well the behaviour of the classifier on a particular dataset.

1.2 Motivation and Objectives

Since PPI plays important role in many biological processes occurring in cells such as cellular motion, gene regulation and transduction, many researchers are working to understand these different biological functions based on protein sequence or their structure. Researchers are conducting their research work either by following labour intensive experimental or computational approaches. Thus, understanding these biological functions, helps understand diseases and develop drugs for their cure. Valuable knowledge generated by these computational methods about these interactions greatly helps biological research.

During their life span, proteins interact with each other or even within themselves to change their shape or other complexes to perform a specific biological function. Previously labour intensive approaches were available such as affinity chromatography or co-immunoprecipitation. These days, high throughput methods such as tandem affinity purification, mass spectroscopy, yeast two hybrid, bimolecular fluorescence complementation are available. But these methods are not applicable to all the proteins in all the organisms as they suffer from some system errors [1, 16, 40]. Therefore, new computational approaches are attracting more researchers for studying protein-protein interactions due to the fact that these are cost effective. I propose a computational model, which I call PPIEE (protein-protein interaction using electrostatic energy), to predict and analyze protein interaction types (obligate and non-obligate) using electrostatic energies as properties. Thus, studying these two types of PPIs will help researchers gather more information for understanding and explaining biological processes and mechanisms for complex formation. It may also help in drug development for curing diseases related to PPI.

Different features and prediction methods can be used to solve this specific problem. In this thesis, I use electrostatic energies as properties. Using these properties, I show that it is

better than previously used properties such as NOXclass features (interface area, interface area ratio, amino acid composition of the interface, correlation between amino acid compositions of interface and protein surface, gap volume index and conservation scores of the interface) [50], desolvation energies [4], and others. With the proposed properties, I also implement a model grouping by amino acids or by atom types for prediction purposes.

1.3 Problem Statement

In this thesis, I study the problem of predicting and analyzing protein-protein interaction types (obligate and non-obligate) using electrostatic energies as properties. The aim is to distinguish between two types of complexes, i.e. obligate and non-obligate, as mentioned above. It is a very important problem in the field of proteomics.

1.4 Hypothesis

The thought for using electrostatic energies as properties in the proposed PPIEE is due to the fact that electrostatic interactions are long ranged. These may go upto 10Å or even more. Thus, they cover wider interface area yielding more knowledge about atoms present in the interface. There has been a lot of research in field of electrostatic energies. But the idea of using these energies as properties to predict the protein interactions types (obligate and non-obligate) is innovative and a framework has been designed to evaluate and validate this hypothesis.

1.5 Contributions

I focus on prediction of two types of protein-protein interactions, namely obligate and non-obligate to evaluate the results efficiently. The main contributions in this thesis are:

- To calculate the electrostatic energies for the Zhu dataset [50] and the Mintseris dataset [32] using computer tools knowns as PDB2PQR [11] and APBS [5].
- To propose PPIEE that uses electrostatic energies as properties to predict protein-protein interaction types.
- To integrate data from four different file formats, namely PDB, PQR, OUT and INTERFACE.
- The use of electrostatic energies on these datasets with different grouping criteria such as by atom types and amino acids.
- Visual analysis using electrostatic potential for some complexes in order to analyze interactions from a different perspective.

In summary, I propose PPIEE to solve this type of problems efficiently and effectively. The development of an automatic tool is important, which downloads the structural information from PDB [7], and calculates the pairs of atoms in the interface for a particular threshold distance and then, computes their electrostatic energies. Then, it arranges the data in the form of features so that classification can be applied to predict the interaction types.

1.6 Thesis Organization

The thesis is organized in six chapters. Chapter II provides a survey of obligate and non-obligate PPI and the prediction methods used to determine those types. Chapter III presents different feature generation methods that can be used for prediction. Chapter IV describes the proposed PPIEE model and features, and all required methods for the experiments. Chapter V discusses the experimental results with the proposed approach and comparisons with some of the existing methods. Finally, Chapter VI concludes the thesis with some discussions and identifies the problems arising from this work and relevant future work.

Chapter 2

Obligate and Non-obligate PPI Prediction

2.1 Proteins

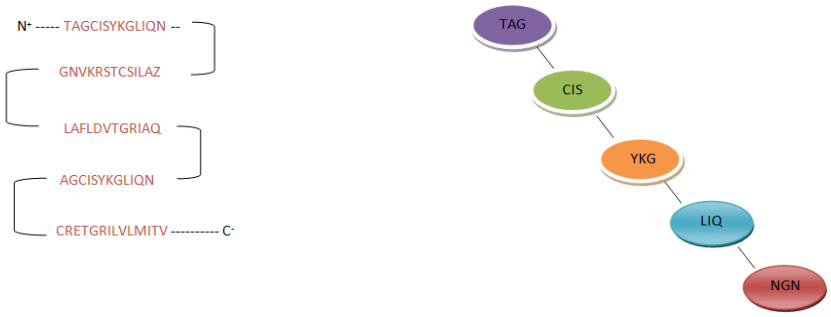
In 1838, Dutch chemist Gerhardus Johannes Mulder first described proteins, which were named by Swedish chemist J*ö*ns Jakob Berzelius [46]. Proteins are the most important class of biochemical molecules, although lipids and carbohydrates are also essential for life. Proteins are the basis for the major structural components of human and animal tissue. Proteins are natural polymer molecules consisting of amino acids. The number of amino acids in a protein may range from a few to several thousands. In general, a protein is an organic compound, made of an arranged chain of amino acids which forms a globular or fibrous form [46].

2.2 Protein Structures

A protein is a polymer of amino acids that has four levels of structure [38]:

1. Primary
2. Secondary
3. Tertiary
4. Quaternary

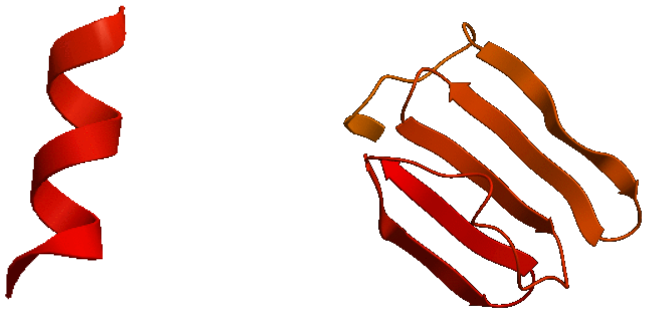
Figure 2.1 shows the primary structure of a typical protein which is made up of a sequence of amino acids along their backbone. Figure 2.1 (a) shows the linear sequence of amino-acids that starts from amino terminal (N) end to the carboxyl-terminal (C) end. Counting the residues always starting at the N terminal ((NH_2) group), which is the place in which the amino group is not involved in the peptide bond. The sequence of the protein is unique to that protein and also defines its structure and function. Figure 2.1 (b) shows a representation of primary structure of a portion of the protein in a different format. It shows some of the 20 standard amino-acids such as alanine, proline, valine etc, in the three letter code format. The structure is held together by peptide bonds which take place during the process of protein biosynthesis.



(a) Primary structure (b) List of amino acids forming primary structure

Figure 2.1: Primary structure of a typical protein- a linear sequence of amino acids beginning from amino-terminal N to carboxyl-terminal C.

The secondary structure as shown in Figure 2.2 consists alpha helices, beta sheets and turns. These were suggested in 1951 by Linus Pauling and are defined by patterns of hydrogen bonds among the main chain peptide groups.



(a) Alpha helix (b) Beta sheet

Figure 2.2: Representation of the secondary structure of PDB ID 1ava, a non-obligate complex, showing an alpha helix and a beta sheet which are the main components. The figure was generated with the help of ICM browser [48].

Figure 2.2(a) represents an alpha helix in which it is coiled like a loosely coiled spring. The coiling occurs in clockwise direction as it goes away from us. The main criteria for the alpha helix is that the amino acid side chain should cover and protect the backbone (H-bonds). Attractive forces between different atoms cause formation of specific structures. Figure 2.2(b) shows beta-pleated sheets in which chains are folded so that they lie along with each other. These sheets are actually anti-parallel in nature. The folded chains are held together by hydrogen bonds that makes them more stable in nature. The figures were generated with the help of ICM browser [48] by entering complex PDB ID 1ava, which is obtained from the Zhu dataset [50].

Tertiary structure is the three-dimensional structure of a single protein molecule. The alpha helices and beta sheets are folded into globular form. This folding is mainly due to the hydrophobic interactions. Figure 2.3 shows the tertiary structure of PDB ID 1dev, chain A. It is a combination of alpha helices as ribbons and beta sheets as arrow heads. It is made up of several domains and each domain has its own function. The bits of the protein chain which are just random coils or loops are shown as bits of strings. These structures are diverse and more complex and largely determined by the biomolecule's primary structure or the sequence of amino acids. Its amino acids are capable of performing diverse functions ranging from molecular recognition to catalysis. The figure was generated with the help of ICM browser [48].

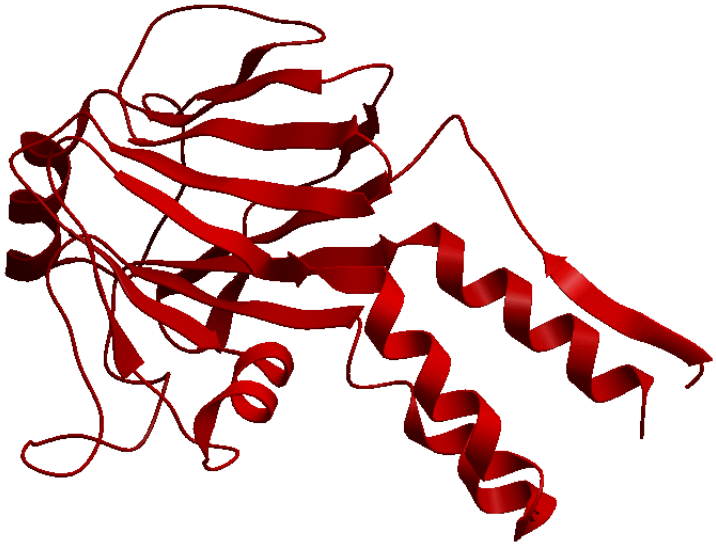


Figure 2.3: Tertiary structure as one sub-unit of PDB ID of 1dev (Chain A), a non-obligate complex. The structure is a combination of alpha helices shown as ribbons and beta sheets shown as arrow heads.

Quaternary structure is the arrangement of multiple folded proteins in a multi-subunit complex. The quaternary structure of a protein can be determined by experimental techniques by placing a sample protein in different experimental positions. Most of the proteins in the human body are found in this state. Examples of proteins in that state include DNA polymerase, hemoglobin and ion channels.

Figure 2.4 shows an example of a protein in its quaternary structure. The four different colors represents four different chains for 1a3n, i.e., Chains A, B, C and D. These sub-units yield a stable structure. This structure involves the clustering of several individual protein chains into a specific shape. PDB ID 1a3n (hemoglobin) is a type of Globular quaternary protein which is clumped into a ball shape as shown. Other examples are insulin and most of the enzymes. This structure may fall apart in a high salt environment. This figure was generated with the help of ICM browser [48].

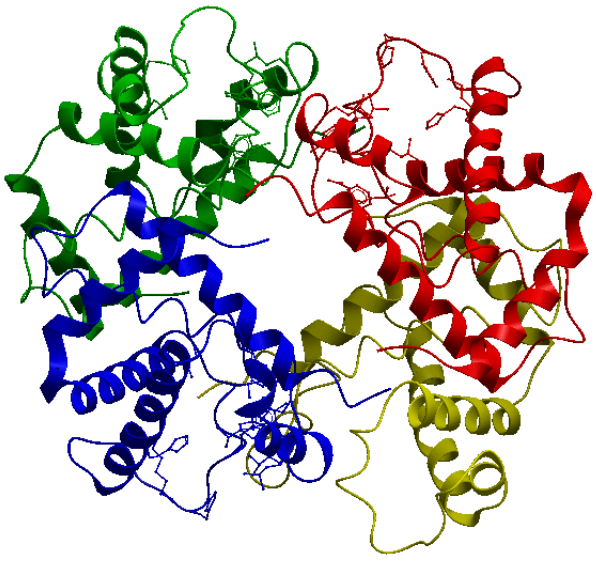


Figure 2.4: Quaternary structure of PDB ID 1a3n, hemoglobin, consisting of Chains A, B, C and D respectively. The four different chains are represented in different colors for clarity.

2.3 Protein-protein Interactions

To perform biological functions, one or more proteins may bind to each other or to DNA or RNA. This is the main reason why protein interaction takes place inside human body. Protein-protein interaction involves [33]:

- Direct contact association of molecules, which means that different molecules belonging to specific amino acids within a protein may interact with each other if they are close to each other under a certain threshold distance. Generally, for direct association molecules, they should be within 7Å distance from each other [8].

- However, if I consider the role of electrostatic interactions, these are considered to be long ranged interactions through the surrounding neighbourhood. Interactions may

take place, if the molecules are within the distances of 10Å or even more [23].

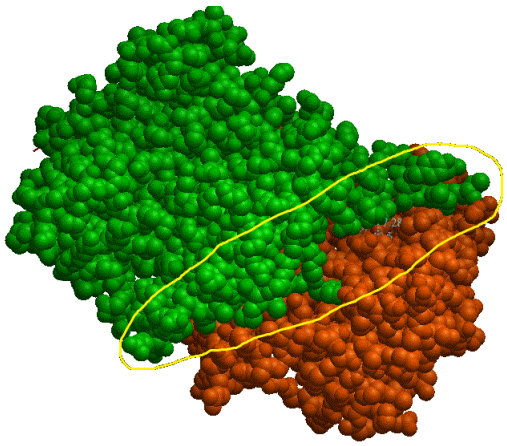


Figure 2.5: Protein-protein interaction for complex PDB ID 1cli. The highlighted area indicates the interface or the place where interaction occurs between the two chains.

Figure 2.5 shows two interacting chains, Chains A and B of PDB ID 1cli. The interacting chains are shown in green and red respectively. The highlighted portion shows the interface where the actual interaction takes place. This figure gives a clear idea how two chains interact with each other, forming a complex. This region is the area of interest, since electrostatic energies for the pairs of atoms present in this region are calculated using available software packages. This figure was generated using ICM browser [48].

There is a need to understand how proteins interact with each other. Since proteins perform different biological processes in a cell including cell growth, nutrient uptake, motility, intercellular communication and apoptosis. Critical aspects required to understand functions of proteins include [17]:

- **Protein sequence and structure** - used to discover motifs that predict protein function.
- **Evolutionary history and conserved sequences** - identifies key regulatory residues.

- **Interactions with other proteins** - functions may be predicted by knowing the function of binding partners.
- **Post translational modifications** - even some changes after the interaction take place.

2.3.1 Domains

The concept of domains was first proposed in 1973 by Wetlaufer [29]. A domain is a three dimensional structure that is part of a protein. It evolves, functions and exists independently from the rest of the protein chain. Domains may vary in length from 25 to 5,000 amino acids. Physical interaction between proteins is also analysed on the basis of the residues of their structural domains. The presence of multiple domains gives rise to great flexibility and mobility. As a result, domain-domain interactions, these properties can be used to predict protein-protein interaction types [29].

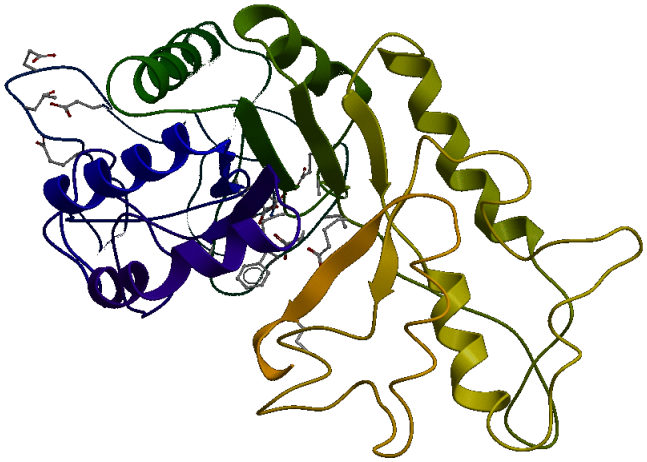


Figure 2.6: Representation of domains in complex PDB ID 1ava (chain A).

Figure 2.6 shows the domains of PDB ID 1ava, which is a non-obligate complex. The

domains were discovered using the Pfam server [39], which represent protein families or domains. The server helps in determining the domains and it carries its architecture. The residues belonging to Chain A beginning from 17 and ending at 324 were found to be a domain and is shown in the figure. This figure was generated using ICM browser [48].

2.3.2 Motifs

A motif refers to a particular amino acid sequence that is characteristic of a particular biological function. An example is the zinc finger motif, which is a functional motif. In other words, it is a set of contiguous secondary structure elements that has specific functional significance or defines a portion of an independently folded domain. Identifying motifs from a sequence is not straightforward since they have functional implications. Some motifs such as the zinc finger motif are easy to identify since they are continuous but other might be discontinuous and might be then difficult to predict. Short linear motifs are short stretches (also known as SLiMs or minimotifs) of a protein sequence that help mediate protein-protein interaction. The main role of SLiMs is to target specific interactions with other protein domains [9].

The sequence logo is a useful graphical representation of the protein motifs. It shows how well amino acids are conserved at each position; the higher the frequency of amino acids, the higher the letter will be, because the better the conservation is at that position. Different amino acids at the same position are scaled according to the information content of that amino acid.

Figure 2.7 shows the sequence logo for the short, linear motif found on the chains of Cytoplasmic Malate Dehydrogenase (PDB ID 4mdh) complex at position 233. The Regular Expression of the SLiM is **[RN][YDG]I[GRW][YF]G**, which also represents the motif *x*-

axis which represents the positions of the SLiM, and the y-axis that reflects the information content of that specific position. The information content of the y-axis is given by (2.1):

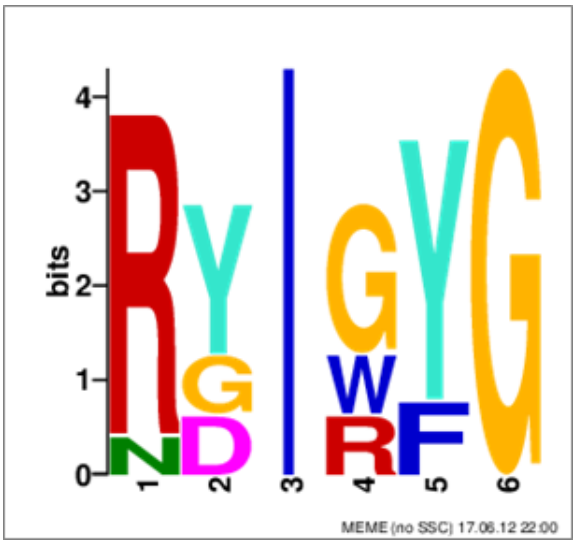


Figure 2.7: One short, linear motif found in both chains of Cytoplasmic Malate Dehydrogenase (PDB ID 4mdh) at position 233. The regular expression of this SLiM is **[RN][YDG]I[GRW][YF]G**.

$$R_i = log_2(20) - (H_i + e_n). \tag{2.1}$$

where H_i is the uncertainty of a position i and H_i is defined as:

$$H_i = -\sum f_{a,i} \log_2 f_{a,i}. \tag{2.2}$$

Here, $f_{a,i}$ is the relative frequency of amino acid a at position i, e_n is the small-sample correction for an alignment of n letters. The height of letter a in column i is given by (2.3):

$$e_n = \frac{(s-1)}{2\ ln(2)\ n}. \tag{2.3}$$

For proteins, the value of s is 20 and n the length of the SLiM.

2.3.3 Protein-protein Interaction types

Another important aspect in studying PPI is to predict different types of complexes, including similarities between subunits (homo/hetero-oligomers), number of subunits involved in the interaction (dimers, trimers, etc.), duration of the interaction (transient vs. permanent) [31], stability of the interaction (non-obligate vs. obligate) [50]. Based on physiological functions, specificity and evolution, PPIs can be divided into four broad categories [33]:

1. Homo and hetero-oligomeric interactions
2. Non-obligate and obligate interactions
3. Transient and permanent interactions
4. Crystal packing and biological interactions

Homo-oligomer and hetero-oligomeric interactions: If the two interacting protein chains of an oligomer have structural symmetry, then those kinds of PPIs are called homo-oligomeric protein-protein interactions. If the two interacting protein chains of an oligomer have differences in their structure, those kinds of PPIs are called hetero-oligomeric protein-protein interactions.

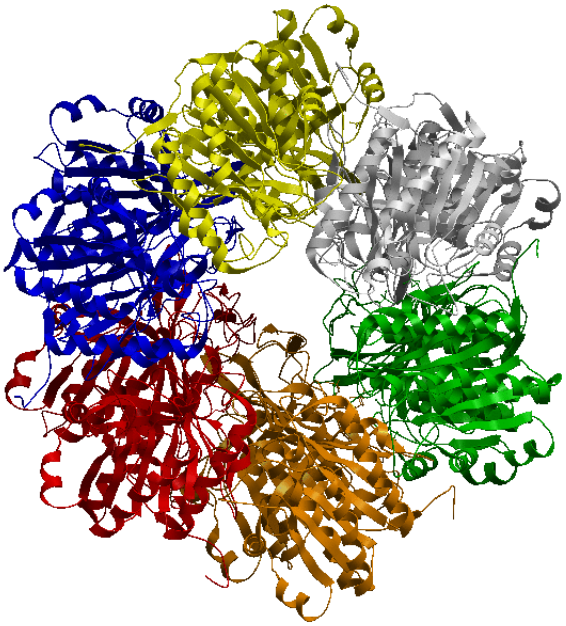


Figure 2.8: Homo-oligomer complex PDB ID 2gbl.

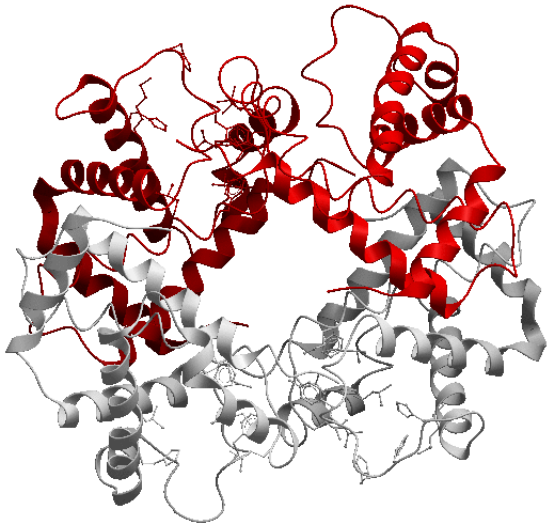


Figure 2.9: Hetero-oligomeric complex PDB ID 1a3n.

Figure 2.8 depicts the quaternary structure of PDB ID 2gbl which is a homo-oligomer complex. This complex contains six identical units, all reflected in different colors which are structurally similar to each other.

Figure 2.9 shows another complex, PBD ID 1a3n, Hemoglobin. It is made up of two homo-dimers as shown in different colors, red and white.

Obligate and Non-obligate interactions: Obligate interactions are usually considered as permanent, while non-obligate interactions can be either permanent or transient [33]. Non-obligate and transient interactions are more difficult to study and to understand due to their instability and short life, while obligate and permanent interactions last for a longer period of time, and hence are more stable [22].

Transient and permanent protein-protein interactions: These kinds of interactions are differentiated on the basis of the lifetime of the interacting protein complexes. Permanent interactions are stable interactions, while transient interactions are less stable interactions.

Crystal packing and biological interactions: During the process of crystallization protein complexes form solid crystals [50]. These kinds of interactions do not serve the biological purpose because they are incapable of performing any biological activity. This type of packing is called crystal packing interaction. All other PPIs that occur perform some biological functions and are called biological PPI.

PPIs can take place between identical or non-identical chains which are based on their structural similarity. If the interacting chains of an oligomer has structural symmetry, then it is called homo-oligomeric PPI, otherwise it is called hetero-oligomeric PPI. While considering the life time of a complex, it can be a transient and permanent PPI. Permanent PPIs are stable complexes, while transient PPIs are less stable as they tend to continue chang-

ing their shapes until they dissociate or result in permanent complexes. To find out about the biological functions performed by different types of complexes, it is important to know about the PPIs.

2.4 Protein-protein Interaction Prediction Approaches

Many researchers worked on predicting protein-protein interaction types with different approaches. These approaches are mainly divided into two main categories, namely:

1. Experimental Approaches
2. Computational Approaches

Traditionally, prediction of PPI was limited to experimental techniques such as co-immunoprecipitation or affinity chromatography [1, 16, 40]. These methods are labor-intensive and often the results of these approaches contain systematic errors. Since the data for prediction is growing, these experimental processes become less applicable. Computational approaches are more popular these days because they are less expensive to use. Computational approaches can be further divided into two classes.

1. Methods based on evolutionary conservation information
2. Those solely structure based on geometrical and physiochemical properties of the proteins.

Modern multiple sequence alignment algorithms including residue substitution matrices allow detection of conservation signal. One type of prediction depends on the algorithm for deriving scores from the alignment [34]. But the main problem is that many protein interfaces are not conserved at all. Alignment-independent prediction methods rely on the

assumption that protein interfaces are quite different from the rest of the surface, since they vary in their geometrical and physiochemical properties from rest of the surface. The physical properties of the interface are highly diverse [36]. To decide between which approach is better depends upon the nature of the protein of interest and available resources.
To predict obligate and non-obligate PPIs, many classifiers including random forest (RF), Bayes, decision trees, logistic regression, neural networks, SVM and LDR have been used. Different classifiers achieve different performances based on the type of data and properties used. Some research in [12, 15, 41] achieved a classification accuracy of 70% for the problem of distinguishing obligate and non-obligate interactions with a wide range of parameters and different types of features such as desolvation energy, amino acid composition, conservation scores, electrostatic energies, and hydrophobicity.

Descriptors based on the 3D structure of proteins used for the prediction are as follows [24]:

1. **Neighbour list**: Residues in the spacial vicinity to the residue in question. Generally, 9-20 residues are taken into consideration.

2. β **Factor**: It is the crystallographic measure that approximates the flexibility of the residue.

3. **Secondary structure**: This structure represents alpha helices, beta sheets and loops.

4. **Sequence distance**: The separation between residues within the same patch. Some results shows that structurally contiguous residues are more likely to form interaction sites.

Descriptors based on the evolutionary features used for prediction are as follows:

1. **Sequence profile**: Extracted from multiple sequence alignment, profiles reveal patterns of evolutionary conservation.

2. **Conservation Score**: A quantification of the level of conservation of an individual position.

3. **Conservation of physiochemical traits**: If the position is not conserved, scoring conservation of traits such as charge, hydrophobicity, or size may improve prediction.

Descriptors based on physiochemical properties:

1. **Hydrophobicity**: It is the physical property of a molecule that is repelled from mass of water. Hydrophobic molecules often cluster together in water to form micelles. Several different scales are available.

2. **Electrostatic potential**: It co-relates with the dipole moment, electro-negativity and partial charges. In other words, it is the potential energy of the protons at a particular location near a molecule. It requires the 3D structure of the protein.

3. **Atom propensities**: It serves as a way to sum physiochemical properties across residues in a patch.

4. **Desolvation energies**: Used in prediction of protein interaction types and rigid-body docking.

The proposed PPIEE model uses LDR and SVM methods for classification. PPIEE uses LDR coupled with Bayesian both linear and quadratic. The details of these classifiers are explained in Chapter 3.

Chapter 3

Feature Extraction and Prediction

3.1 Pattern Recognition

Pattern recognition is the scientific field whose goal is clustering, feature selection and extraction and classification of objects into a number of categories or classes [45]. An example of pattern recognition is classification whose aim is to assign labels to unknown objects. A simple example can be whether to determine that a given email is spam or non-spam. Generally speaking, the objects can be protein sequences, signal waveforms or images. There are several application areas of pattern recognition in today's world. It is the most important part for the machine learning systems built for decision making. Typical applications are automatic speech recognition, classification of text in several categories and automatic recognition of images of human faces.

In machine learning and pattern recognition, one of the key factors is to include and select the right features for successful prediction. These are the observed properties of each sample that are used for prediction. The value of the features are usually numeric, but other types such as strings and graphs are also used.

3.1.1 Classifiers

In pattern recognition, classification is the procedure for identifying unknown samples on the basis of training sets of known samples. The classification problem is known as supervised learning in pattern recognition. In supervised learning, the training set is provided with labels attached to each sample. Then, the classifier is trained on these samples. After the classifier is well trained, it can classify unknown samples. On the other hand, unsupervised learning assumes that training data do not have labels attached to the samples and attempts to find the inherent patterns in the data that can hopefully determine the correct output value for new unknown instances. There are various classifiers available [14]:

1. Linear classifiers
2. Quadratic classifiers
3. Support vector machine
4. Decision tree
5. Neural Networks
6. Hidden Markov Models

Among these, I use SVM, linear classifiers and quadratic classifiers for prediction of obligate and non-obligate types of interactions.

3.2 Features Used for PPI Prediction

There are many properties that can be used for PPI prediction. Some of them are [35]:

β-factor : It is the flexibility of the protein complexes during the interaction.

Solvent Accessibility : It is the exposed surface area that affects contact of atoms during the interaction.

Geometric features : The shape index, planarity or curvedness of the interacting complexes can also be used as features.

Evolutionary features : They include conservation scores or sequence profiling information.

Physicochemical features : They include hydrophobicity, electrostatic and desolvation energies.

3.2.1 Tools Used to Calculate Electrostatic Energies

Electrostatic interactions are important in understanding intermolecular interactions. These interactions control the structure and binding of biomolecules. Electrostatics are important, since they are long ranged interactions and because of their influence in charged molecules [5]. This motivated me to focus on electrostatic energies and hence use them as properties for predicting interaction types.

In order to compute electrostatic energies, software packages including PDB2PQR [11] and APBS [6] were used. PDB2PQR was written by Todd Dolinsky while working with Nathan Baker at Washington University in St. Louis, USA. PDB2PQR is a tool that automates common tasks for preparing structures for electrostatic calculations. Its purpose is to add missing heavy atoms, place missing hydrogen atoms and assign charge and radii to PDB files. The output of this package is a PQR structure file, which is the input for APBS. PDB2PQR can either be run through web-servers or installed on a local machine so that

the software can be run with customized parameters. The command line argument which executes the Python script is in (3.1).

$$\$ \text{ python pdb2pqr.py [options] –ff= \{forcefield\} \{path\} \{output path\}} \tag{3.1}$$

This software provides many parameters which can be customized as per needs which are as follows:

- **forcefield**: Currently this software supports AMBER, CHARMM, PARSE and TYL06 forcefields.
- **path**: It requires the input path to the PDB complex. The software will execute this PDB file to output PQR file format.
- **output-path**: It requires the desired name of the PQR file. The file will be saved on the path or location specified here in this parameter.
- **chain name(optional)**: It keeps the chain ID in the output of the PQR file format.
- **–asign-only(optional)**: It only assigns charge and radii. It does not add atoms, or debump, or optimize.
- **–noopt(optional)**: It does not perform hydrogen bonding network optimization.
- **–apbs-input(optional)**: It creates the template APBS input file based on the generated PQR file. It further has some parameters that can be changed to run APBS.
- **–help(optional)**: It displays the usage information.

APBS is a software package used for calculating electrostatic energies for interactions between solutes in salty and aqueous media [6]. It solves the Poisson-Boltzmann equation numerically and evaluates electrostatic calculations ranging from tens to millions of atoms. It basically uses Fekt (Finite element toolkit) which solves the Poison-Boltzmann equation for electrostatic calculations. It is used for modelling bio-molecular solvation through solution. This software was written by Nathan Baker in collaboration with J.Andrew McCammon and Michael Holst. APBS is also integrated with other programs such as Jmol, which help visualize the complexes for better understanding.

APBS can be run either through the web-server or it can be installed on a local machine. The advantage of installing it and running it on a local machine is that it provides the freedom to use more parameters. For PPIEE, electrostatic energies per atom were calculated using APBS. The command line used to run APBS is in (3.2).

$$
\begin{aligned}
&for\ i\ in * .in;\ \ do \\
&\text{echo "Running \$i..."} \\
&(\text{apbs \$}i\ 2{>}\&1)\ \|\ \text{tee \$}\{i\%.in\}.out \\
&\text{done}
\end{aligned}
\tag{3.2}
$$

Some of the parameters which need to be taken care of while running APBS are in the *elec* section as follows:

- **Calcenergy**: Changing the parameter Calcenergy "total" to Calcenergy "comps". The software produces the electrostatic energies per atom.
- **cglen**: It specifies the length of the coarse-grid for multi-grid dimensions.

3.3 Proposed Features on Electrostatic Energies

I consider the interaction between the i^{th} atom of the first interacting chain and the j^{th} atom of the second interacting chain. Then, I calculate the distance between all possible atom pairs. If an atom pair is separated by a distance less than or equal to a certain threshold, that pair is considered to be in the interface of that complex [23]. In most studies, inter-atom distances are not greater than 7Å. But in order to achieve an in-depth analysis for electrostatic energies as properties, in this work, the threshold was varied from 7Å to 13Å. It is to be noticed that even the atoms that are under the surface of proteins pose electrostatic forces towards the stability of the protein complex.

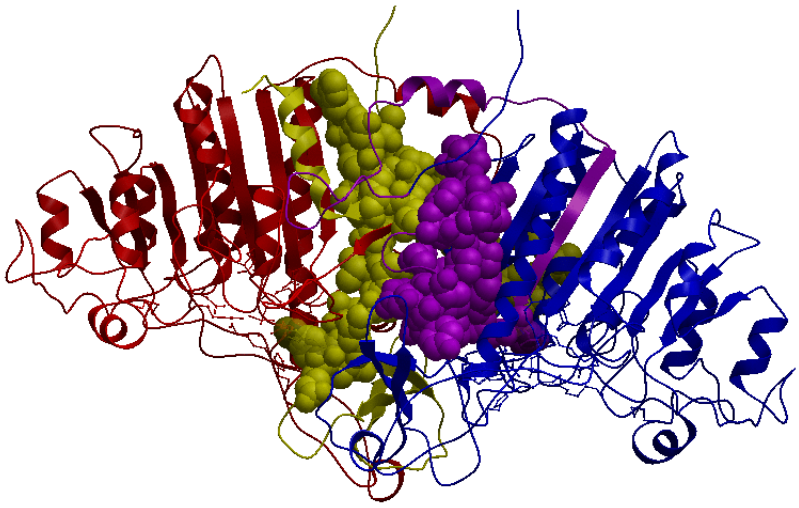


Figure 3.1: An obligate complex PDB ID 1b8j with its interacting chains A and B as shown in different colors, red and blue. This figure was generated with the help of ICM browser. [48].

Figure 3.1 depicts an obligate complex, PDB ID 1b8j, along with its interacting chains A and B shown in different colors, red and blue. Atoms that are under certain threshold

of distance act as interface atoms and are represented as yellow and purple spheres respectively. An atom pair is considered to be in the interface if atoms are under certain threshold distance apart from each other for this complex. Since electrostatic interactions are long ranged, a larger interface area is observed.

3.4 Feature Generation

For each complex in the datasets, structural data from PDB [7] were collected. Two interacting chains were extracted from each PDB file. I consider 18 different atom types as in [49]. Thus, for each protein complex a feature vector with ${}^{18}C_2$+18 = 171 values were obtained, where each feature contains the cumulative average of electrostatic energies for all pairs of atom of the same type in the interface. Since the order of interacting atoms in the pairs is not important, the final length of the feature vector for each complex is 171 which corresponds to the number of unique pairs [42]. I have also considered pairs of amino acids, and for this, I computed ${}^{20}C_2$+20 = 210 values for all pairs of amino acids and by accumulating the electrostatic values of the corresponding atoms for each pair of amino acids. The final length of the feature vector for each complex is 210 which corresponds to the number of unique pairs for 20 standard amino acids. This is how I obtain feature vectors for both atom types (171) and amino acid types (210). Averages for electrostatic values for the pairs of atom types and amino acids are calculated as follows:

$$\Sigma(e_i + e_j)/2. \tag{3.3}$$

where e_i stands for the electrostatic value for the i^{th} interface atom from the first interacting chain and e_j stands for the j^{th} electrostatic value for the interface atom from the second

interacting chain.

Using Equation (3.3), for each complex in the datasets, the averages of electrostatic values were computed for the pairs of atoms on the interface. There are 18 unique type of atoms and 20 amino acids. Table 3.1 [49] is an 18×18 matrix, where all possible combinations of 18 unique atom types are represented. If the interacting chains of a complex are known, the very first task is to find the interacting atoms from both chains under a certain threshold value. Then, the conversion for all the interface atoms to their corresponding unique types is done. The method of conversion is derived from the conversion table that is discussed in [49]. Based on PPIEE experiments and the study of [49], Tables 3.2 and 3.3 are used for atom type conversion. In these conversion tables, the first row of each amino acid contains the original atoms that are inside them and the second row contains the converted unique atom-types. When the unique types are determined, I simply find that pair in the table and obtain the unique types for the atoms. In order to find the interface atoms, I find the Euclidian distance between the two atoms from two interacting chains, which is to be computed from their structural data that can be found in PDB files [7].

Table 3.1 shows an example for the atom-type table. The averaged electrostatic values for the pairs of atoms are stored in their respective positions as shown in the table. With the help of Table 3.2 [49], I was able to determine a unique type for every atom in the interface. Just as an example, if I know that atom Nitrogen belongs to amino acid Alanine, then its unique type will be Nitrogen.

3.4.1 Classifiers Used in This Work

3.4.1.1 Bayesian Decision Theory

Bayesian decision theory is a fundamental statistical approach for pattern classification. It assumes that decision problems are based on probabilistic terms. The main focus of this work is about the discriminant function for normal densities. The expression for the multivariate normal distribution is [13]:

$$g_i(\mathbf{x}) = -\tfrac{1}{2}(\,\mathbf{x}-\boldsymbol{\mu}_i)^t\,\textstyle\sum_i^{-1}\,(\,\mathbf{x}-\boldsymbol{\mu}_i\,) - \tfrac{d}{2}\,\ln\,2\pi - \tfrac{1}{2}\,\ln\,|\textstyle\sum_i|^d + \ln\,P(\,\omega_i\,). \tag{3.4}$$

where $\mathbf{x}$ is a $d-$ component column vector, $\boldsymbol{\mu}$ is a $d-$ component mean vector and Σ is the $d \times d$ covariance matrix, which is always symmetric and positive semidefinite, and $P(\omega_i)$ are the prior probabilities.

Some special cases for the discriminant functions are the following:

Case 1: This case is when the features are statistically independent and have the same variance, σ^2. Geometrically, this case is when samples fall on equal-sized clusters. The discriminant function is:

$$g_i(\mathbf{x}) = -\tfrac{1}{2\sigma^2}[\mathbf{x}^t\,\mathbf{x} - 2\boldsymbol{\mu}_i^t\,\mathbf{x} + \boldsymbol{\mu}_i^t\boldsymbol{\mu}_i] + \ln\,P(\omega_i). \tag{3.5}$$

Case 2: When the covariance matrices for all the classes are identical but otherwise arbitrary. Geometrically, in this situation all the samples fall on hyper ellipsoidal clusters of equal size and shape [13]. This is what is known as linear Bayesian classifier (LB).

$$g_i(\mathbf{x}) = -\tfrac{1}{2}(\,\mathbf{x}-\boldsymbol{\mu}_i)^t\;\Sigma^{-1}\;(\,\mathbf{x}-\boldsymbol{\mu}_i)+\;\ln\;P(\omega_i). \tag{3.6}$$

where for two classes Σ_i can be obtained as:

$$\Sigma = \;\tfrac{1}{2}[\Sigma_1\;+\;\Sigma_2]. \tag{3.7}$$

In (3.6) $P(\omega_i)$ is the prior probability of class ω_i. To classify each feature vector $\mathbf{x}$, one measures the squared Mahalanobis distance from each $\mathbf{x}$ to each of the c mean vectors and assign $\mathbf{x}$ to the category of the nearest mean.

Case 3: It is the general multivariate normal case, in which the covariance matrices are different for each category. This is called quadratic Bayesian classifier (QB). The classification function is as follows [13]:

$$g_i(\mathbf{x}) = \;\mathbf{x}^t\;\mathbf{W}_i\;\mathbf{x}+\mathbf{w}_i^t\;\mathbf{x}+\omega_{i0}. \tag{3.8}$$

where $\mathbf{W}_i$ is:

$$\mathbf{W}_i = -\tfrac{1}{2}\;\Sigma_i^{-1}. \tag{3.9}$$

and $\mathbf{w}_i$ is:

$$\mathbf{w}_i = \sum_i^{-1} \boldsymbol{\mu}_i. \tag{3.10}$$

3.4.1.2 Linear Dimensionality Reduction

The basic idea of LDR is to represent an object of dimension n (171 or 210) as a lower-dimensional vector of dimension d, achieving this by performing a linear transformation. Consider two classes, obligate as ω_1 and non-obligate as ω_2, represented by two normally distributed random vectors $\mathbf{x}_1 \sim N(\mathbf{m}_1, \mathbf{S}_1)$ and $\mathbf{x}_2 \sim N(\mathbf{m}_2, \mathbf{S}_2)$, respectively, with p_1 and p_2 the *a priori* probabilities. After the LDR is applied, two new random vectors $\mathbf{y}_1 = \mathbf{A}\mathbf{x}_1$ and $\mathbf{y}_2 = \mathbf{A}\mathbf{x}_2$, where $\mathbf{y}_1 \sim N(\mathbf{A}\mathbf{m}_1;\ \mathbf{A}\mathbf{S}_1\mathbf{A}^t)$ and $\mathbf{y}_2 \sim N(\mathbf{A}\mathbf{m}_2;\ \mathbf{A}\mathbf{S}_2\mathbf{A}^t)$ with $\mathbf{m}_i$ and $\mathbf{S}_i$ being the mean vectors and covariance matrices in the original space, respectively. The aim of LDR is to find a linear transformation matrix $\mathbf{A}$ in such a way that the new classes ($\mathbf{y}_i = \mathbf{A}\mathbf{x}_i$) are as separable as possible. Let $\mathbf{S}_W = p_1\mathbf{S}_1 + p_2\mathbf{S}_2$ and $\mathbf{S}_E = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^t$ be the within-class and between-class scatter matrices respectively. Various criteria have been proposed to measure separability between the classes [41]. I focus on the following three LDR methods:

Fisher's discriminant analysis (FDA) [12, 15] : Its optimization criterion is as follows:

$$J_{FDA}(\mathbf{A}) = tr\left\{(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1}(\mathbf{A}\mathbf{S}_E\mathbf{A}^t)\right\} \tag{3.11}$$

The matrix $\mathbf{A}$ is found by considering the eigenvector corresponding to the largest

eigenvalue of $\mathbf{S}_{FDA} = \mathbf{S}_W^{-1}\mathbf{S}_E$.

Heteroscedastic discriminant analysis (HDA) [28] : It aims to obtain the matrix $\mathbf{A}$ that maximizes the function:

$$J_{HDA}(\mathbf{A}) = tr\left\{(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1}\left[\mathbf{A}\mathbf{S}_E\mathbf{A}^t - \mathbf{A}\mathbf{S}_W^{\frac{1}{2}}\frac{p_1\log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_1\mathbf{S}_W^{-\frac{1}{2}})+p_2\log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_2\mathbf{S}_W^{-\frac{1}{2}})}{p_1p_2}\mathbf{S}_W^{\frac{1}{2}}\mathbf{A}^t\right]\right\} \tag{3.12}$$

This criterion is maximized by obtaining the eigenvectors, corresponding to the largest eigenvalues, of the matrix:

$$\mathbf{S}_{HDA} = \mathbf{S}_W^{-1}\left[\mathbf{S}_E - \mathbf{S}_W^{\frac{1}{2}}\frac{p_1\log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_1\mathbf{S}_W^{-\frac{1}{2}})+p_2\log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_2\mathbf{S}_W^{-\frac{1}{2}})}{p_1p_2}\mathbf{S}_W^{\frac{1}{2}}\right] \tag{3.13}$$

Chernoff discriminant analysis (CDA) [41] : It aims to maximize the following function:

$$J_{CDA}(\mathbf{A}) = tr\{p_1p_2\mathbf{A}\mathbf{S}_E\mathbf{A}^t(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1} + \log(\mathbf{A}\mathbf{S}_W\mathbf{A}^t) - p_1\log(\mathbf{A}\mathbf{S}_1\mathbf{A}^t) - p_2\log(\mathbf{A}\mathbf{S}_2\mathbf{A}^t)\} \tag{3.14}$$

In this work I use the gradient-based algorithm proposed in [41], which maximizes the function in an iterative way. For this gradient algorithm, a learning rate, α_k needs to be computed. In order to ensure that the gradient algorithm converges, α_k is maximized by using the secant method. One of the keys in this algorithm is the random initialization of the matrix $\mathbf{A}$, and in this work, ten different initializations were performed, choosing the solution for $\mathbf{A}$ that gives the maximum Chernoff distance.

3.4.2 Support Vector Machine

I also use SVM, which is widely used in bioinformatics for classification. It takes the set of input vectors and predicts the possible classes of output based on the support vectors [19]. The kernel trick allows one to map the original vectors onto a much higher dimensional space that hopefully makes class separation easier. Basically, there are two kinds of margins in SVM. The first one is called hard margin and the second one is called soft margin. In the hard margin SVM, the training data is linearly separable in the input space while in soft margin data can be either be linearly or non-linearly separable. If the training data is not linearly separable they can be mapped onto a higher dimension feature space to increase separability.

3.4.2.1 Hard-Margin Support Vector Machine

Let $X_i(i = 1,....n)$ be d-dimensional training input vectors which belong to class 1 or class 2 and the corresponding labels be $y_i = 1$ for class 1 and $y_i = -1$ for class 2. For linearly separable data the the decision function is as follows [3]:

$D(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$

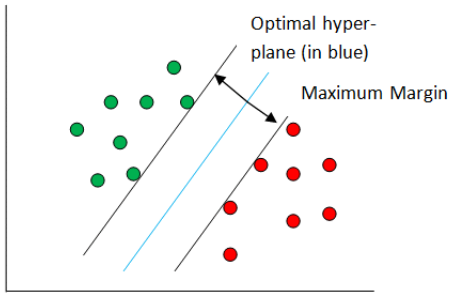


Figure 3.2: Optimal separating hyperplane in the two dimensional space.

Figure 3.2 shows the optimal hyperplane which best classifies between two classes. As shown in the figure, the hyperplane in blue has the maximum margin with respect to both classes and is the called *optimal separating plane*.

The optimal separating hyperplane can be obtained by minimizing :

$Q(\mathbf{w}) = 1/2 \parallel \mathbf{w} \parallel^2$

with respect to $\mathbf{w}$ and b subject to the constraints:

$y_i(\mathbf{w}^T \mathbf{x}_i + b) >= 1$ for $i = 1, ...M$

The square of the Euclidean norm $\parallel \mathbf{w} \parallel$ is to make the optimization problem solvable by quadratic programming. The data is linearly separable, if there exists $\mathbf{w}$ and b that satisfy the above equation. In order to map the input vectors to a higher dimensional feature space, with infinite dimensions, the above mentioned constraints are converted into the unconstrained problem [3]:

$$\mathbf{Q}(\mathbf{w}, \mathbf{b}, \alpha) = \tfrac{1}{2}\mathbf{W}^T\mathbf{w} - \sum_{i=1}^{n} \alpha_i[\mathbf{y}_i(\mathbf{w}^T)\mathbf{x}_i + b) - 1]. \tag{3.15}$$

where $\alpha = \alpha_1, ...\alpha_m$ are the non-negative Lagrange multipliers. Then, the maximization problem is as follows:

$$\mathbf{Q}(\alpha) = \sum_{i=1}^{m} \alpha_i \; - \; \frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j. \tag{3.16}$$

with respect to a_i subject to the constraints:

$$\sum_{i=1}^{M} y_i \alpha_i \; = \; 0 \tag{3.17}$$

where $\alpha_i \geq 0$ for $i = 1, ...M$

All data associated with a_i are known as support vectors and from the standard point of precision of calculations, the average is calculated among the support vectors which is:

$$b \; = \; \frac{1}{\|S\|} \sum_{i \in S} (y_i - \mathbf{w}^T \mathbf{x}_i) \tag{3.18}$$

Now the unknown datum **x** can be classified into:

$$\begin{aligned} &Class\ 1\ if\ D(\mathbf{x}) \; > \; 0, \\ &Class\ 2\ if\ D(\mathbf{x}) \; < 0 \end{aligned} \tag{3.19}$$

This function classifies the data into two different classes. In special case $D(\mathbf{x}) = 0$, the sample is said to be unclassifiable.

3.4.2.2 Soft-Margin Support Vector Machines

Soft margin can also be used for the linearly separable data. But in case the data are non-linearly separable and there is no feasible solution. The aim is to find an optimal hyperplane

that aims to maximize the distance between the classes. Thus, following criteria is used [3]:

$$\mathbf{Q}(\mathbf{w},\mathbf{b},\beta) = \tfrac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{i=1}^{M} \beta_i^p. \tag{3.20}$$

where C has to be optimized.

3.4.2.3 The Kernel Trick

In SVM, the aim of the hyperplane is to maximize the separation between classes. But if the training data are not linearly separable then the optimal hyperplane is not good enough. Thus, to increase the separability, the original input vectors are mapped onto a higher dimensional dot product space called *feature space*. For this, there is a mapping function $\phi(\mathbf{x})$ that maps the $\mathbf{x}$ into the dot-product feature space and $g(\mathbf{x})$ satisfies [3]:

$$H(\mathbf{x},\mathbf{x}') \; = \; \phi^t(\mathbf{x})\, \phi(\mathbf{x}') \tag{3.21}$$

Using the kernel trick, the problem in the feature space is as follows:

$$\mathbf{Q}(\mathbf{a}) = \sum_{i=1}^{M} \alpha_i \; - \; \tfrac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j y_i y_j H(\mathbf{x}_i,\mathbf{x}_j) \tag{3.22}$$

and the problem is solved without explicitly mapping onto the feature space provided that the kernel satisfies Mercer's condition in (3.23). This whole procedure is called *kernel trick*.

$$\sum_{i,j=1}^{M} h_i \, h_j \, H(\mathbf{x}_i,\mathbf{x}_j) \; \geq 0 \tag{3.23}$$

Some kernels which are used in the SVM are as follows:

- Linear Kernel:

If a classification problem is linearly separable, then there is no need to map the data onto higher dimensions. Thus, the linear kernel is as follows:

$$H(\mathbf{x},\mathbf{x}') \;=\; \mathbf{x}^t\mathbf{x}' \tag{3.24}$$

- Polynomial kernel:

  The polynomial kernel with degree d, where d is a natural number, is given by:

$$H(\mathbf{x},\mathbf{x}') \;=\; (\mathbf{x}^t\mathbf{x}'+1)^d \tag{3.25}$$

  Here one is added with degree equal to less than d are included.

- Radial Basis Function (RBF) kernels:

  The RBF kernel is given by:

$$H(\mathbf{x},\mathbf{x}') \;=\; exp(-\gamma \parallel \mathbf{x}-\mathbf{x}' \parallel^2 \tag{3.26}$$

  where γ is a positive parameter for controlling the radius, which has to be optimized.

3.4.3 Prediction Evaluation

Cross validation plays an important role in validating prediction results. It is a technique for assessing how the results of statistical analysis will generalize to unknown samples of any dataset. It involves partitioning of sample data into subsets. The most common approach used is the K fold cross validation. PPIEE uses 10-fold cross validation in classification. Cross validation provides a framework for creating several train/test sets assuring that each data point is in the test set at least once. In this framework, the data is split into 10 equal

sized groups. For each iteration, one of the group acts as test set while the remaining ones are training sets. Each iteration is called a **fold**. The model is trained on the training set and then evaluated on the test set. These iterations are repeated ten times and thus is known as 10-fold cross validation. The total values for *TP*, *TN*, *FP* and *FN* are calculated to compute the accuracies as stated in (3.27).

To compute the accuracy for each classifier, the following equation is used:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.27}$$

- ***TP*** is when positive is classified as positive
- ***TN*** is when negative is classified as negative
- ***FP*** is when negative is classified as positive
- ***FN*** is when positive is classified as negative

TP is the number of correctly identified obligate complexes and *TN* is the number of correctly identified non-obligate complexes. *FP* and *FN* are number of incorrectly classified obligate and non-obligate complexes respectively.

Table 3.1 shows the fixed value of actual energy for each atom pair for the threshold of 5Å. Table 3.2 and Table 3.3 shows designation of the 18 unique atom types [49]. With the help of this table, PPIEE is able to determine the unique type for each interface atom.

Table 3.1: Atom type table.

	N	CA	C	O	GCA	CB	KNZ	KCD	DOD	RNH	NND	RNE	SOG	HNE	YCZ	FCZ	LCD	CSG
N	-0.724	-0.903	-0.722	-0.322	-0.331	-0.603	1.147	0.937	0.752	0.502	0.405	0.495	-0.116	-0.092	-0.412	-0.499	-1.005	-2.06
CA	-0.903	-0.842	-0.85	-0.332	-0.365	-0.531	1.139	0.918	0.852	0.452	0.445	0.436	-0.044	-0.165	-0.539	-0.59	-1.123	-2.028
C	-0.722	-0.85	-0.704	-0.371	-0.308	-0.499	1.14	0.846	0.788	0.49	0.408	0.418	-0.045	-0.089	-0.415	-0.478	-0.963	-2.033
O	-0.322	-0.332	-0.371	-0.016	0.205	-0.059	1.414	1.075	1.152	0.831	0.758	0.649	0.383	0.241	-0.073	-0.166	-0.65	-1.65
GCA	-0.331	-0.365	-0.308	0.205	0.182	-0.009	1.502	1.385	1.192	0.912	0.872	0.943	0.434	0.392	-0.062	-0.021	-0.342	-1.212
CB	-0.603	-0.531	-0.499	-0.059	-0.009	-0.469	1.31	1.01	0.859	0.671	0.591	0.565	0.122	0	-0.438	-0.533	-1.034	-1.8
KNZ	1.147	1.139	1.14	1.414	1.502	1.31	3.018	2.911	1.157	2.635	1.848	2.699	1.587	1.557	1.004	1.34	1.252	0.7
KCD	0.937	0.918	0.846	1.075	1.385	1.01	2.911	2.811	0.989	2.439	1.622	2.525	1.361	1.277	0.685	0.938	0.814	0.411
DOD	0.752	0.852	0.788	1.152	1.192	0.859	1.157	0.989	1.978	0.695	1.386	0.641	1.002	0.642	0.618	0.894	0.792	-0.029
RNH	0.502	0.452	0.49	0.831	0.912	0.671	2.635	2.439	0.695	1.589	1.395	1.515	1.022	0.948	0.457	0.707	0.586	0.021
NND	0.405	0.445	0.408	0.758	0.872	0.591	1.848	1.622	1.386	1.395	1.3	1.283	0.931	0.829	0.484	0.633	0.389	-0.046
RNE	0.495	0.436	0.418	0.649	0.943	0.565	2.699	2.525	0.641	1.515	1.283	1.309	0.921	0.777	0.265	0.484	0.274	-0.253
SOG	-0.116	-0.044	-0.045	0.383	0.434	0.122	1.587	1.361	1.002	1.022	0.931	0.921	0.559	0.319	0.301	0.212	-0.155	-0.949
HNE	-0.092	-0.165	-0.089	0.241	0.392	0	1.557	1.277	0.642	0.948	0.829	0.777	0.319	-0.301	-0.159	-0.132	-0.338	-1.324
YCZ	-0.412	-0.539	-0.415	-0.073	-0.062	-0.438	1.004	0.685	0.618	0.457	0.484	0.265	0.301	-0.159	-0.314	-0.478	-0.964	-1.41
FCZ	-0.499	-0.59	-0.478	-0.166	-0.021	-0.533	1.34	0.938	0.894	0.707	0.633	0.484	0.212	-0.132	-0.478	-0.687	-1.24	-1.784
LCD	-1.005	-1.123	-0.963	-0.65	-0.342	-1.034	1.252	0.814	0.792	0.586	0.389	0.274	-0.155	-0.338	-0.964	-1.24	-1.873	-2.402
CSG	-2.06	-2.028	-2.033	-1.65	-1.212	-1.8	0.7	0.411	-0.029	0.021	-0.046	-0.253	-0.949	-1.324	-1.41	-1.784	-2.402	-3.742

Table 3.2: Atom conversion table (a).

Amino acid	Conversion												
ALA	N	CA	CB	C	O	OXT							
	N	CA	CB	C	O	O							
ARG	N	CA	CB	CG	CD	NE	CZ	NH1	NH2	C	O	OXT	
	N	CA	CB	FCZ	RNE	RNE	RNH	RNH	RNH	C	O	O	
ASN	N	CA	CB	CG	OD1	ND2	C	O	OXT				
	N	CA	CB	NND	NND	NND	C	O	O				
ASP	N	CA	CB	CG	OD1	OD2	C	O	OXT				
	N	CA	CB	DOD	DOD	DOD	C	O	O				
CYS	N	CA	CB	SG	C	O	OXT						
	N	CA	CB	CSG	C	O	O						
GLN	N	CA	CB	CG	CD	OE1	NE2	C	O	OXT			
	N	CA	CB	FCZ	NND	NND	NND	C	O	O			
GLU	N	CA	CB	CG	CD	OE1	OE2	C	O	OXT			
	N	CA	CB	FCZ	DOD	DOD	DOD	C	O	O			
GLY	N	CA	C	O	OXT								
	N	GCA	C	O	O								
HIS	N	CA	CB	CG	ND1	CD2	NE2	CE1	C	O	OXT		
	N	CA	CB	HNE	HNE	HNE	HNE	HNE	C	O	O		
ILE	N	CA	CB	CG2	CG1	CD	C	O	OXT	CD1			
	N	CA	CB	LCD	FCZ	LCD	C	O	O	LCD			

Table 3.3: Atom conversion table (b).

Amino acid	Conversion															
LEU	N	CA	CB	CG	CD1	CD2	C	O	OXT							
	N	CA	CB	FCZ	LCD	LCD	C	O	O							
LYS	N	CA	CB	CG	CD	CE	NZ	C	O	OXT						
	N	CA	CB	FCZ	KCD	KNZ	KNZ	C	O	O						
MET	N	CA	CB	CG	SD	CE	C	O	OXT							
	N	CA	CB	FCZ	FCZ	LCD	C	O	O							
PHE	N	CA	CB	CG	CD1	CD2	CE1	CE2	CZ	C	O	OXT				
	N	CA	CB	FCZ	FCZ	FCZ	FCZ	FCZ	FCZ	C	O	O				
PRO	N	CA	CB	CG	CD	C	O	OXT								
	N	CA	CB	CB	CB	C	O	O								
SER	N	CA	CB	OG	C	O	OXT									
	N	CA	SOG	SOG	C	O	O									
THR	N	CA	CB	OG1	CG2	C	O	OXT								
	N	CA	CB	SOG	FCZ	C	O	O								
TRP	N	CA	CB	CG	CD2	CE2	CE3	CD1	NE1	CZ2	CZ3	CH2	C	O	OXT	
	N	CA	CB	FCZ	FCZ	FCZ	FCZ	FCZ	HNE	FCZ	FCZ	FCZ	C	O	O	
TYR	N	CA	CB	CG	CD1	CE1	CD2	CE2	CZ	OH	C	O	OXT			
	N	CA	CB	FCZ	FCZ	YCZ	FCZ	YCZ	YCZ	SOG	C	O	O			
VAL	N	CA	CB	CG1	CG2	C	O	OXT								
	N	CA	CB	LCD	LCD	C	O	O								

Chapter 4

Methodology

4.1 The Proposed Model

To predict obligate and non-obligate protein complexes with good accuracy, I follow the PPIEE model as depicted in Figure 4.1. PPIEE helps in predicting the types of interactions. In order to make it time efficient, the algorithms have been modified with improved and modified search techniques. These steps can be easily understood with the data flow diagram (DFD) depicted in Figure 4.1.

The description of the whole procedure is as follows:

Step 1: (Processing datasets):

The datasets were obtained from Mintseris *et al.* [32] and Zhu *et al.* [50]. Details of datasets are explained in Section 4.2.

Step 2: (Downloading structural information of each complex from PDB server [7]):

All the files are downloaded from the PDB server. PDB is a repository for the 3D struc-

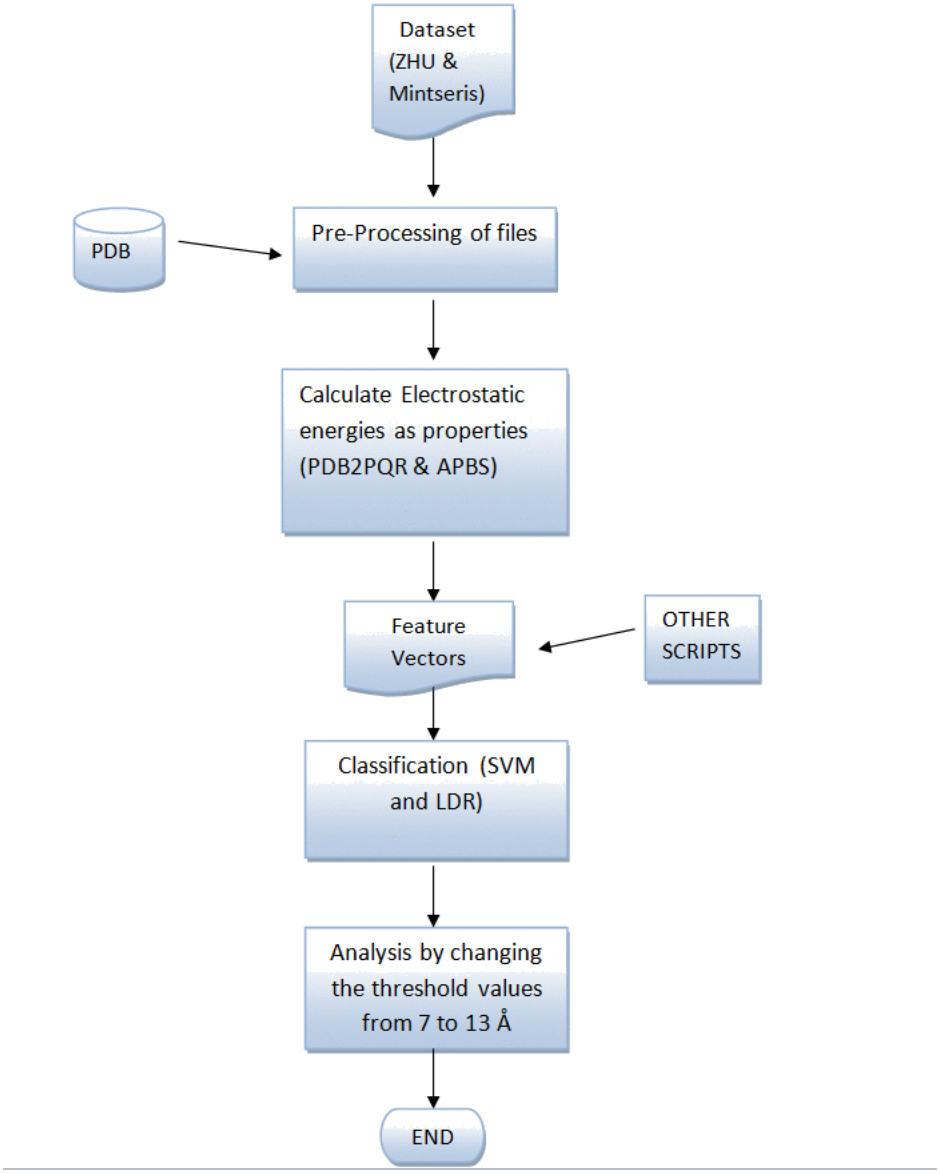


Figure 4.1: Proposed PPIEE model for prediction of protein interaction types.

tural data of large biological molecules, such as proteins and DNA. The data are generally obtained by X-ray crystallography or NMR spectroscopy submitted by biologists or biochemists from around the world. All the files downloaded from this server are saved in PDB format. The data are freely accessible on the internet via the PDB web-

site, http://www.rcsb.org/pdb/home/home.do .

Step 3: (Preprocessing of PDB files):

In this step, pre-processing of all PDB files is done. Only the lines that begin with the keyword "ATOM" are used in the pre-precessed files. These lines are useful because of the information they contain, such as atom number, atom name, residue number, residue name, atomic coordinates for each atom and occupancy factor. The tools PDB2PQR and APBS including all other pre-processing scripts run on these PDB files. All other lines are removed, since they are not required. The files are split up into two files on the basis of the interacting chains names in order to reduce processing time for experiments.

Step 4: (Initialization of the properties):

Running PDB2PQR [11] installed on a local machine converts PDB files into PQR format which act as input for APBS. Then, running APBS [6] installed on a local machine generates the output files that have electrostatic energies for each atom in the interface. Details are explained in Chapter 3, Section 3.2.1.

Step 5: (Calculate the feature vector with electrostatic energies):

For each complex, all atom pairs in the interface are generated using their atomic coordinates given in their respective PDB file format. Based on their atomic coordinates, Euclidean distance is calculated between all possible pair of atoms. If a specific atom pair distance is less than or equal to a certain threshold value, say, 10$\mathring{A}$, then those atom pairs are considered to be in the interface and stored in the interface file. Then, using the atom type conversion table, the unique type for each atom is determined [49]. In the next step,

feature vectors are obtained which contains the cumulative average of electrostatic energies for all pairs of atoms in the interface as explained in Chapter 3, Section 3.3. Thus, a dataset is obtained, that is ready for the next classification step.

Step 6: (Classification using LDR and SVM):

LDR (HDA, FDA, CDA) combined QB and LB classifiers [12] and SVM are then applied to the datasets obtained in the previous step, to achieve the best possible accuracies for prediction of protein interaction types.

Step 6: (Post analysis):

Some other experiments are performed by changing the threshold values which are the distances calculated between atom pairs from interacting chains, ranging from 7Å to 13Å were done. The details are shown in Chapter 5, Section 5.1.3. Also, visual analysis using electrostatic potential for some complexes is done in order to yield an in-depth analysis. The analysis is done for an obligate complex PDB ID 2min and non-obligate complex PDB ID 1a2k as shown in Chapter 5, Section 5.1.4.

4.2 Datasets

This work is based on two well-known datasets, referred to as the MW dataset [32] and the ZH dataset [50] which contains obligate and non-obligate complex names with the corresponding chain information. All the complexes in the ZH dataset [50] have the characteristics that one chain is interacting with only one chain, while in the MW dataset [32] there are complexes that have more than two chains in the interaction. Zhu *et al.* obtained obligate interactions from the literature while non-obligate complexes were taken from the set of non-obligate interactions and transient interactions [50].

The datasets used for the experiments are as follows: The table 4.1 shows the two datasets used for the experiments.

Table 4.1: Dataset description.

Dataset name	Reference	No. of obligate complexes	No. of non-obligate complexes
MW	Mintseris *et al.* [32]	115	212
ZH	Zhu *et al.* [50]	75	62

The conventions used in Table 4.2 are ZH-AT which stands for the ZH dataset, atom type, while ZH-AA stands for the ZH dataset, amino acid type. Similarly, MW-AT stands for the MW dataset, atom type while MW-AA stands for the MW dataset, amino acid type.

Table 4.2: Description of the datasets used in this study.

Datasets	Reference	Atom type	Amino acid type
MW	Mintseris *et al.* [31]	MW-AT	MW-AA
ZH	Zhu *et al.* [50]	ZH-AT	ZH-AA

Table 4.3 contains all non-obligate complexes of the MW dataset while Table 4.4 contains all the obligate complexes of the same dataset. Table 4.5 shows all obligate complexes

for the ZH dataset while Table 4.6 shows all non-obligate complexes for the same dataset.

Table 4.3: Non-obligate complexes in the MW dataset.

1wq1 G:R	1icf AB:I	1fle E:I	2prg B:C	1h59 A:B	1fbv A:C	1c4z A:D	1fns A:HL
1tmq A:B	1bkd R:S	1evt A:C	3c98 A:B	1xdt R:T	1t7p A:B	1zbd A:B	1ahw AB:C
1fbi HL:X	1ar1 AB:CD	1qfw AB:IM	1i85 B:D	1osp HL:O	1fsk A:BC	1kxt A:B	2jel HL:P
1bqh AB:GM	1jma A:B	1kac A:B	1hez AB:E	1sbb A:B	1e6j HL:P	1wej F:HL	1bgx HL:T
1akj AB:DE	1qfu AB:HL	2hmi AB:CD	1i9r ABC:HL	1a14 HL:N	1bzq A:L	1nsn HL:S	1mr1 A:D
1lk3 A:HL	1ezv E:XY	1iqd AB:C	1dee CD:GH	1ao7 ABC:DE	1gc1 C:G	1bj1 HL:VW	1k4c AB:C
1f51 AB:E	1bdj A:B	1eay A:C	1kmi Y:Z	7cei A:B	1clv A:I	1ava A:C	1dhk A:B
1bvn P:T	1i1a AB:CD	1l6x A:B	1qkz A:HL	1t83 AB:C	1f34 A:B	1dpj A:B	1im3 AB:D
1g73 AB:C	3g94 A:BC	1fak HL:T	1jw9 B:D	1jtg A:B	1jtd A:B	1d5x A:C	1i4e A:B
1ib1 AB:E	1kyo O:W	2mta A:C	2pcc A:B	1f3v A:B	1eja A:B	1lpb A:B	1dtd A:B
1eer A:B	1ibr A:B	1i7w A:B	1f60 A:B	1itb A:B	1ay7 A:B	1dx5 AM:I	1kkl ABC:H
4sgb E:I	1dev A:B	1l0o AB:C	1smf E:I	1a2k AB:C	1dfj E:I	1avg HL:I	1k90 A:D
1g4y B:R	1jch A:B	1ebd AB:C	1e6e A:B	1gaq A:B	1f80 A:E	1buv M:T	4htc HL:I
1stf E:I	2tec E:I	1acb E:I	1e96 A:B	1qav A:B	1f02 I:T	1tab E:I	2ptc E:I
1gl1 A:I	1ezx AB:C	1toc AB:R	1cgi E:I	1eai A:C	1avx A:B	1azz A:CD	1bml A:C
2btc E:I	1gh6 A:B	1iod AB:G	1agr A:E	1avz B:C	1rlb ABCD:E	1aro L:P	1awc A:B
1fqj A:C	1ak4 A:D	1i3o ABCD:E	1kxp A:D	1d4x A:G	2btf A:P	1hx1 A:B	1atn A:D
1dkg AB:D	1fq1 A:B	1fin A:B	1b6c A:B	1bi8 A:B	1buh A:B	1qo0 A:DE	1ugh E:I
1df9 B:C	1jiw I:P	1f93 AB:EF	1noc A:B	1es7 AC:B	1k5d A:C	1hwg A:BC	1fg9 AB:C
1ebp A:CD	1du3 A:DEF	1cmx A:B	1euv A:B	1he1 A:C	1kcg AB:C	1efx ABC:D	1de4 CF:A
1m10 A:B	1ghq A:B	1flt VW:X	1gxd A:C	1kzy A:C	1ycs A:B	1gla F:G	1cxz A:B
2sic E:I	1jsu AB:C	1lb1 A:B	1is8 AB:KL	1doa A:B	2mta A:HL	1b9y AB:C	4cpa I:0
1gp2 A:B	1g0y I:R	1ijk A:BC	1i4d AB:D	1k5d A:B	1n2c AB:EF	1mah A:F	1gcq B:C
1www VW:X	1i2m A:B	1kgy A:E	1c1y A:B	1gl4 A:B	1d2z A:B	3ygs C:P	1grn A:B
1cs4 AB:C	1ki1 A:B	1efu A:B	3sgb E:I	1fqv A:B	1k3z AB:D	1m4u A:L	1m2o AC:B
1mbu A:C	1fc2 C:D	1ml0 A:D	1gvn AC:B	1o6s A:B	1h2k A:S	1m1e A:B	1o94 AB:CD
1nf5 A:B	1gzs A:B	1nbf A:D					

Table 4.4: Obligate complexes in the MW dataset.

1nbw AC:B	2q33 A:B	1mjg AB:M	1h32 A:B	1m2v A:B	1mro A:B	1jk8 A:B	1exb A:E
1dxt A:B	1cpc A:B	1f3u A:B	1poi A:B	1hcn A:B	1prc C:HLM	1jb7 A:B	1e8o A:B
1b8m A:B	1raf A:BD	2min A:B	1jk0 A:B	1jb0 AB:D	1kfu L:S	1req A:B	1hxm A:B
1k8k C:G	1vkx A:B	1spp A:B	1jro A:BD	2ahj A:B	1li1 AB:C	1qgw A:C	1e9z A:B
2kau B:C	1jmz AG:B	1ktd A:B	1ytf BC:D	3gtu A:B	1g8k A:B	1vcb A:B	1hr6 AE:B
1h8e A:D	1fxw A:F	1gka A:B	3pce A:M	1kqf B:C	1kqf A:B	1jnr A:B	1dj7 A:B
1b7y A:B	1jb0 AB:E	1ihf A:B	1efv A:B	1dce A:B	1hzz AB:C	1luc A:B	1lti AC:DEHFG
4rub AD:T	1ir1 A:S	1a6d A:B	1jwh A:CD	1k8k A:B	1hfe L:S	1jb0 C:D	1jb0 C:E
1jb0 AB:C	1h2r L:S	1tbg A:E	2mta H:L	1ezv D:H	1ezv C:F	1be3 CDEGK:A	1ldj A:B
1ffv A:B	1sgf A:BY	1ffu A:C	1jkj A:B	2kau A:C	1eex A:G	1eg9 A:B	1fcd A:C
1k8k C:F	1k8k D:F	1k8k B:F	1c3o A:B	1ep3 A:B	1h4i A:B	1k28 A:D	1k3u A:B
1e50 A:B	1jv2 A:B	1dm0 A:BCFDE	1k8k A:E	1jmx A:G	1l9j C:HLM	1dii A:C	1l7v AB:C
1ld8 A:B	1e6v A:B	1mro A:C	1mro B:C	1fm0 D:E	1qlb B:C	2bs2 A:B	1aui A:B
1b4u A:B	1hsa A:B	1dtw A:B	1gpw A:B	1qdl A:B	1ccw A:B	1eex A:B	1go3 E:F
1dkf A:B	1h2v C:Z	1fs0 E:G					

Tables 4.7 and 4.8 contain the list of complexes from the ZH and MW datasets and their corresponding types, obligate (O) or non-obligate (NO), for which APBS failed to compute the electrostatic energies per atom due to some internal errors in the software package.

Table 4.5: Obligate complexes in the ZH dataset.

4mdh A:B	3tmk A:B	2utg A:B	2pfl A:B	2nac A:B	2hhm A:B	2hdh A:B	1gux A:B
2ae2 A:B	1yve I:J	1ypi A:B	1xso A:B	1xik A:B	1wgj A:B	1vok A:B	1efv A:B
1vlt A:B	1trk A:B	1spu A:B	1sox A:B	1smt A:B	1qu7 A:B	1qor A:B	1qfh A:B
1qfe A:B	1qbi A:B	1qax A:B	1qae A:B	1pp2 L:R	1one A:B	1nse A:B	1msp A:B
1kpe A:B	1jkm A:B	1isa A:B	1hss A:B	1hjr A:C	1hgx A:B	1gpe A:B	1f6y A:B
1dor A:B	1cp2 A:B	1coz A:B	1cnz A:B	1cmb A:B	1cli A:B	1c7n A:B	1byk A:B
1byf A:B	1brm A:B	1bo1 A:B	1bjn A:B	1b9m A:B	1b8j A:B	1b8a A:B	1b7b A:C
1b5e A:B	1b3a A:B	1at3 A:B	1aq6 A:B	1aom A:B	1ajs A:B	1aj8 A:B	1afw A:B
1a4i A:B	1a0f A:B	2aai A:B	1tco A:B	1req A:B	1pnk A:B	1luc A:B	1dce A:B
1ffv A:B	1sgf A:BY	1ffu A:C	1jkj A:B	2kau A:C	1eex A:G	1eg9 A:B	1fcd A:C
1k8k C:F	1k8k D:F	1k8k B:F	1c3o A:B	1ep3 A:B	1h4i A:B	1k28 A:D	1k3u A:B
1e50 A:B	1jv2 A:B	1dm0 A:BCFDE	1k8k A:E	1jmx A:G	1l9j C:HLM	1dii A:C	1l7v AB:C
1ld8 A:B	1e6v A:B	1mro A:C	1mro B:C	1fm0 D:E	1qlb B:C	2bs2 A:B	1aui A:B
1b4u A:B	1hsa A:B	1dtw A:B	1gpw A:B	1qdl A:B	1ccw A:B	1eex A:B	1go3 E:F
1dkf A:B	1h2v C:Z	1fs0 E:G	1h2a L:S	1b34 A:B	1ahj A:B		

Table 4.6: Non-obligate complexes in the ZH dataset.

1a4y A:B	1tmq A:B	1dn1 A:B	1lfd A:B	1jtd A:B	1eth A:B	1hlu A:P	1qbk B:C
1uea A:B	1emv A:B	1stf E:I	1f60 A:B	2pcb A:B	1bml A:C	1cmx A:B	1aro L:P
1cvs A:C	1ycs A:B	3hhr A:B	1frv A:B	1avz B:C	1eg9 A:B	1cc0 A:E	1rrp A:B
1wq1 R:G	1bi7 A:B	1dhk A:B	1fin A:B	1cqi A:B	1d09 A:B	1ak4 A:D	1zbd A:B
1c0f S:A	1tx4 A:B	1pdk A:B	1kac A:B	1i8l A:C	1i2m A:B	1euv A:B	1dow A:B
1buh A:B	1bkd R:S	1b6c A:B	1atn A:D	1agr E:A	4sgb I:E	2sic I:E	2ptc I:E
1tgs I:Z	1tab I:E	1smp I:A	1kxq H:A	1gla F:G	1fss A:B	1f34 A:B	1eai C:A
1cse I:E	1bvn T:P	1avw A:B	1aqv A:B	1ava A:C			

These complexes are not included in the experiments.

Table 4.7: List of complexes from the ZH dataset with their corresponding types obligate (O) or non-obligate (NO), for which the electrostatic energies could not be computed.

PDB ID	Complex type
1cc0 A:E	NO
1qbk B:C	NO
1b8a A:B	O
1cli A:B	O
1qav A:B	NO
1bkd R:S	NO

Table 4.8: List of complexes from the MW dataset with its corresponding types obligate (O) or non-obligate (NO), for which the electrostatic energies could not be computed.

PDB ID	Complex type
1b7y A:B	O
1be3 CDEGK:A	O
1jb0 AB:C	O
1jb0 AB:D	O
1jb0 AB:E	O
1jro A:BD	O
1jv2 A:B	O
1k28 A:D	O
1kqf A:B	O
1ldj A:B	O
1m2v A:B	O
1mjg AB:M	O
1nbw AC:B	O
1prc C:HLM	O
1bgx HL:T	NO
1de4 CF:A	NO
1ezv E:XY	NO
1is8 ABEJCIDHGF:KLOMN	NO
1m2o AC:B	NO
1o94 AB:CD	NO
1qfu AB:HL	NO
2hmi AB:CD	NO
4cpa I:J	NO
2q33 A:B	O

4.3 Pre-processing Algorithim

This algorithm was designed to speed up the process the experiments. The algorithm works for the correspondence between the following files:

1) INTERFACE file

2) PDB file

3) PQR file

4) OUT file

The main purpose of Algorithm 1 is to traverse the four files as stated above and extract the electrostatic values for the interface atoms. The pairs of atoms that are present on the interface are listed in the INTERFACE file and their corresponding electrostatic values are present in the OUT file. For all atoms present in the interface, the algorithm obtains their corresponding residue names, residue numbers and chain names from the PDB file. Then, it matches four values (atom name, residue names, residue numbers and chain names) from the PDB file to the PQR file and obtain the corresponding PQR atom number. From this PQR atom number, it retrieves the right electrostatic values from the OUT file. Algorithm 1 uses the Binary Search algorithm and the extended version of Binary Search called SearchPQR, that makes the search faster. SearchPQR algorithm is an extended version of Binary Search. It performs binary search but within a specific range of values. This, in turn, makes the algorithm faster. Also, it significantly reduces the processing time for all the experiments performed in the PPIEE model.

Algorithm 2 is a simple Binary Search algorithm. It establishes the correspondence between PQR and OUT files. This algorithm was implemented to search atom numbers in OUT files, so that the corresponding electrostatic values are retrieved and stored in a particular format. These values help make the feature vectors for classification purposes.

Algorithm 3 works for the correspondence between PDB and PQR files, since it involves searching residue numbers from PDB files to PQR files. This algorithm also involves binary search. Even this algorithm takes care of repeated numbers while searching and is able to search within a certain range, which makes the search very fast for all the samples in the dataset. It is the modified version of Algorithm 2. Because of this, there is a significant

reduction in processing time.

4.4 Complexity of the Pre-processing Algorithm

Before implementing the binary search, the order of the original algorithm was cubic. The reason for this cubic complexity was due to the fact that the searching operation was done between three different files, PDB, PQR and OUT files. Suppose that the length of the PDB, PQR and OUT files are k,l and m respectively. For each atom in the interface, I need to find out its electrostatic energy. In order to retrieve the values, the entire processing takes $k \times l \times m$ iterations searching the full length of three mentioned files. Then, if k = l = m =n, the time complexity would be $O(n^3)$. It consumes a lot of computing time and resources. To save time and resources, a binary search algorithm was implemented. If the original number of items is k, then after the first iteration $k/2$ items are remaining and then $k/4$ and then $k/8$ and so on. So this will at most take $log_2(k) + 1$ iterations and worst case is $O(log(k))$. After implementing this binary search, the entire search between the files is in a single loop since all the files are referred in the same loop calling the binary search function. This makes the complexity of Algorithm 1 to be $O(k\ log(k))$ where k is the length of the input. The input here is the length of the file in which the search operation takes place.

Algorithm 1 Main Algorithm for correspondence between files.

For $interfaceatoms_1$
for $i \leftarrow 1, N$ $(Interface\ atoms_1)$ **do**
  $IA_1 \leftarrow Interfaceatom[i].atomnum_1$
  if $EVtable[IA_1] == 50000$ **then**
    $AtomNoPDB_1 \leftarrow BinarySearchPDB(AI_1, PDB_1)$
    $ResNoPDB_1 \leftarrow PDB_1\ [AtomNoPDB_1]$
    $AtomNamePDB_1 \leftarrow PDB_1\ [AtomNoPDB_1]$
    $AtomNoPQR_1 \leftarrow SearchPQR(ResNoPDB_1, AtomNoPDB_1)$
    $AtomNoAPBS_1 \leftarrow [AtomNoPQR_1 - 1]$
    if $AtomNoAPBS_1 == AtomNoPQR_1 - 1$ **then**
      $ElecValue_1 = APBSEV_1[AtomNoPQR_1 - 1].elecvalue()$
    else
      $ElecValue_1 \leftarrow BinarySearchAPBSEV(AtomNoPQR_1 - 1)$
    end if
  else
    $ElecValue_1 \leftarrow EVtable[IA_1]$
  end if

For $interfaceatoms_2$
  for $j \leftarrow 1, N$ $(Interface\ atoms_2)$ **do**
    $IA_2 \leftarrow Interfaceatom[i].atomnum_2$
    if $EVtable[IA_2] == 50000$ **then**
      $AtomNoPDB_2 \leftarrow BinarySearchPDB(AI_2, PDB_2)$
      $ResNoPDB_2 \leftarrow PDB_2\ [AtomNoPDB_2]$
      $AtomNamePDB_2 \leftarrow PDB_2\ [AtomNoPDB_2]$
      $AtomNoPQR_2 \leftarrow SearchPQR(ResNoPDB_2, AtomNoPDB_2)$
      $AtomNoAPBS_2 \leftarrow [AtomNoPQR_2 - 1]$
      if $AtomNoAPBS_2 == AtomNoPQR_2 - 1$ **then**
        $ElecValue_2 = APBSEV_2[AtomNoPQR_2 - 1].elecvalue()$
      else
        $ElecValue_2 \leftarrow BinarySearchAPBSEV(AtomNoPQR_2 - 1)$
      end if
    else
      $ElecValue_2 \leftarrow EVtable[IA_2]$
    end if
  end for
end for

Algorithm 2 Binary Search Algorithm from searching between PDB files and PQR files.

```
BinarySearch(list[], searchTarget)
last ← length(list[])
first ← 1
# While there are elements to be searched
while First = Last do
    middle ← (first + last)/2
# if current middle value is the searchTarget
    if list[middle] = searchTarget then
        return middle
# if current middle value is less than the searchTarget
    else if list[middle] < searchTarget then
        first ← middle + 1
# if current middle value is larger than the searchTarget
    else
        last ← middle − 1
    end if
end while
return 0 if searchTarget not found
```

Algorithm 3 Modified Binary Search for searching between PQR files and OUT files.

```
SearchPQR(list[], searchTarget)
last ← length(list[])
first ← 1
# While there are elements to be searched
while First = Last do
    middle ← (first + last)/2
    Mid1 ← middle − 1
# if current middle value is the searchTarget
    if list[middle] = searchTarget then
        return middle
        V = list(middle)
        while (V = list(middle)) do
            Index[] = middle
            middle = middle + 1
            Index1[] = Mid1
            Mid1 = Mid − 1
        end while
        Sort Index1 and Index
        W[] = Index1 + Index
        return W
# if current middle value is less than the searchTarget
    else if list[middle] < searchTarget then
        first ← middle + 1
# if current middle value is larger than the searchTarget
    else
        last ← middle − 1
    end if
end while
return 0 if searchTarget not found
```

Chapter 5

Results

5.1 Results and Discussion

5.1.1 Experimental Settings

PPIEE has been tested with LDR coupled with a Bayesian classifier and SVM. For both the ZH and MW datasets, I use linear and RBF kernels of SVM. In addition, the parameters have been optimized for the values of C and Gamma on a grid search in order to obtain better accuracies for all datasets used. The SVM is implemented in the Bioinformatics toolbox of Matlab version R2011b which has been used for all the experiments. The details of the code are explained in Appendix A. The computer machine used has Intel i7 processor with 3.40 GHz of clock frequency, with 8GB of RAM. The operating system is 64-bit Windows 7 Professional.

5.1.2 Classification Results and Comparisons

After running the classifiers on the ZH-AT dataset, I observed that the accuracy of SVM is 96.18%. As mentioned earlier, the RBF kernel was also applied with optimized values of C and Gamma obtaining a better accuracy of 97.71%. Considering the ZH dataset for amino acid type using LDR, accuracy is 96.06%. Using SVM and after optimizing the values of C and Gamma, increases accuracy to 96.18%. Similarly, the RBF kernel was also applied to MW datasets with the optimized values of C and Gamma. For the MW-AT dataset, PPIEE obtained a better accuracy of 97.36% while for the MW-AA dataset, the obtained accuracy was 95.38%.

In Table 5.1, I also compare the results with other prediction models using different properties. I compare PPIEE with prediction of obligate and non-obligate using desolvation energies as properties [30]. Their work was based on the ZH and MW datasets. The best accuracy they obtained was 83.21% for the ZH-AT dataset while for PPIEE, the best accuracy is 97.17% on the same dataset with electrostatic energies rather that desolvation energies. All these results are for the threshold value of 10Å, which indicates the maximum distance between atoms to be considered part of the interaction.

Table 5.1: Comparison of PPIEE with desolvation energies as properties for ZH and MW datasets

		Properties (energy)	
Dataset type	Features	Desolvation	Electrostatic
ZH-AA	210	78.39%	96.17%
ZH-AT	171	83.21%	97.17%
MW-AA	210	78.83%	95.38%
MW-AT	171	78.53%	97.36%

In Table 5.2, I compare PPIEE results with different approaches already reported in literature. As shown in the table, Zhu *et al.* predicted obligate and non-obligate interactions

using interface properties and obtained an accuracy of 75.2%. Also, Rueda *et al.* predicted obligate and non-obligate interactions using interface properties with solvent accessible surface area and obtained an accuracy of 81.83%. With PPIEE, I predict obligate and non-obligate interactions using 210 features for electrostatic energies as properties and obtained an accuracy of 96.18% . All these results are for the threshold value of 10Å.

PPIEE results are also compared with the approach of [26] on the BNCP-CS dataset, which consists of 75 obligate interactions and 62 non-obligate interactions. In their approach, they calculate solvent accessible surface area and applied SVM to predict the type of interaction. In their experiments they obtained an accuracy of 92.2%, while in PPIEE experiments using electrostatic energies as properties, the best accuracy obtained is **97.17%**, an increase of about 5% with respect to [26].

Table 5.2: Prediction results and comparison with other approaches and properties on the ZH-AA dataset.

No. of Features	Classifier	Accuracy	Properties used	References
6	SVM	75.2%	Interface	ZH *et al.* [50]
26	LDR	78.27%	Interface	Rueda *et al.* [43]
46	LDR	81.83%	Solvent accessible area and interface area	Rueda *et al.* [43]
210	SVM	92.20%	Solvent accessible area	Liu *et al.* [26]
210	SVM	96.17%	Electrostatic energies	PPIEE

In Table 5.3, I compare PPIEE results on the MW-AA dataset. As shown in the table,

Rueda *et al.* predicted obligate and non-obligate interactions for different number of features using interface properties and obtained an accuracy of 77.54%. Also, Rueda *et al.* predicted interactions using interface properties combined with solvent accessible surface area and obtained an accuracy of 77.25%. With PPIEE, I predict obligate and non-obligate interactions using 210 features for electrostatic energies as properties and obtained an accuracy of 94.57%, implying an increase of more than 15% with respect to [43]. All these results are for the threshold value of 10Å.

Table 5.3: Results of prediction and comparison with other approaches and properties on the MW-AA dataset

No. of Features	Classifier	Accuracy	Properties used	References
4	LDR	77.96%	Interface	Rueda *et al.* [43]
24	LDR	77.54%	Interface	Rueda *et al.* [43]
44	LDR	77.25%	Solvent accessible area and interface area	Rueda *et al.* [43]
210	SVM	95.38%	Electrostatic energies	PPIEE

In Table 5.4, I am comparing PPIEE results using MW-AA and MW-AT datasets respectively. The comparison is with desolvation energy, which is used as properties in [30]. As shown in the table, there is a significant increase in the accuracy of about 18%, which certainly proves electrostatic energy as the best property for predicting protein interaction types. All these results are for the threshold value of 10Å.

Table 5.4: Comparison with the desolvation energy approach for MW-AA and MW-AT datasets.

Dataset	Feature	Accuracy	Properties used	References
MW-AA	210	78.53%	Desolvation energies	Rueda *et al.* [30]
MW-AT	171	78.83%	Desolvation energies	Rueda *et al.* [30]
MW-AA	210	95.38%	Electrostatic energies	Proposed (PPIEE)
MW-AT	171	97.36%	Electrostatic energies	Proposed (PPIEE)

5.1.3 Analysis of Distance Threshold

In order to achieve a better insight of the classification results for ZH and MW datasets, I tried different experiments by varying the threshold values which are the distances calculated between atom pairs of interacting chains, ranging from 7Å to 13Å.

Table 5.5: Classification results for the MW and ZH datasets for a threshold value of 7Å.

		Linear Classifier LDR			Quadratic Classifier LDR		
Datasets	SVM	FDA	HDA	CDA	FDA	HDA	CDA
ZH-AA	95.42	87.79	90.08	95.40	83.21	89.31	94.66
ZH-AT	96.18	71.76	71.76	93.13	69.47	73.28	93.13
MW-AA	90.73	71.19	69.54	83.11	71.85	68.21	67.55
MW-AT	95.03	92.38	82.45	94.37	92.38	82.45	94.04

Table 5.5, shows results for all datasets for a threshold value of 7Å. For ZH-AA the best accuracy obtained is 95.42%. Similarly for ZH-AT, the best accuracy is 96.18%. For MW-AA, the best accuracy is 90.73% while for MW-AT, the best accuracy is 95.03%.

Table 5.6, shows results for all datasets for the threshold value of 8Å. For ZH-AA the best accuracy obtained is 96.95%. Similarly for ZH-AT, the best accuracy is 94.66%. For MW-AA, the best accuracy recorded is 99.67% while for MW-AT, the best accuracy is

Table 5.6: Classification results for MW and ZH datasets for a threshold value of 8Å.

		Linear Classifier LDR			Quadratic Classifier LDR		
Datasets	SVM	FDA	HDA	CDA	FDA	HDA	CDA
ZH-AA	96.95	87.02	89.31	91.60	87.02	87.02	90.08
ZH-AT	94.66	80.15	81.68	94.66	80.92	79.39	95.42
MW-AA	99.66	94.39	90.76	98.02	90.76	95.71	95.71
MW-AT	93.07	93.73	86.47	95.05	92.74	87.46	94.06

93.07%.

Table 5.7: Classification results for MW and ZH datasets for a threshold value of 9Å.

		Linear Classifier LDR			Quadratic Classifier LDR		
Datasets	SVM	FDA	HDA	CDA	FDA	HDA	CDA
ZH-AA	96.18	90.08	90.08	93.13	87.02	90.84	93.89
ZH-AT	94.68	85.88	88.55	95.41	85.50	86.26	95.42
MW-AA	99.67	96.04	88.78	98.02	95.05	95.05	95.71
MW-AT	92.74	90.10	82.51	96.37	89.77	81.52	95.05

Table 5.7, shows results for all datasets for the threshold value of 9Å. For ZH-AA the best accuracy obtained is 96.18%. Similarly for ZH-AT, the best accuracy is 94.68%. For MW-AA, the best accuracy recorded is 99.67% while for MW-AT, the best accuracy is 92.74%.

Table 5.8: Classification results for MW and ZH datasets for a threshold value of 10Å.

		Linear Classifier LDR			Quadratic Classifier LDR		
Datasets	SVM	FDA	HDA	CDA	FDA	HDA	CDA
ZH-AA	96.17	89.31	89.31	95.42	76.34	90.08	95.42
ZH-AT	97.17	93.89	93.89	95.42	88.55	93.89	95.42
MW-AA	95.38	79.21	86.80	94.72	78.88	86.014	89.44
MW-AT	97.36	97.03	86.14	98.35	97.03	88.12	98.68

Table 5.8, shows results for all datasets for the threshold value of 10Å. For ZH-AA the best accuracy obtained is 96.18%. Similarly for ZH-AT, the best accuracy is 97.17%.

For MW-AA, the best accuracy recorded is 95.38% while for MW-AT, the best accuracy is 97.36%.

Table 5.9: Classification results for MW and ZH datasets for a threshold value of 11Å.

		Linear Classifier LDR			Quadratic Classifier LDR		
Datasets	SVM	FDA	HDA	CDA	FDA	HDA	CDA
ZH-AA	96.12	87.60	89.92	95.35	86.05	89.92	93.80
ZH-AT	96.92	90.00	90.00	94.62	87.69	91.54	94.62
MW-AA	94.72	72.87	73.60	84.16	72.61	67.99	84.16
MW-AT	94.06	91.75	82.51	94.72	91.09	83.83	94.02

Table 5.9, shows results for all the datasets for the threshold value of 11Å. For ZH-AA the best accuracy obtained is 96.12%. Similarly for ZH-AT, the best accuracy is 96.92%. For MW-AA, the best accuracy recorded is 94.72% while for MW-AT, the best accuracy is 94.06%.

Table 5.10: Classification results for MW and ZH datasets for a threshold value of 12Å.

		Linear Classifier LDR			Quadratic Classifier LDR		
Datasets	SVM	FDA	HDA	CDA	FDA	HDA	CDA
ZH-AA	96.15	84.62	85.38	93.08	80.77	88.46	89.23
ZH-AT	97.71	76.34	79.39	94.66	73.28	76.34	94.66
MW-AA	94.39	79.54	73.27	87.79	79.21	71.62	81.85
MW-AT	89.11	82.84	87.13	87.79	83.17	88.12	88.12

Table 5.10, shows results for all datasets for the threshold value of 12Å. For ZH-AA the best accuracy obtained is 96.15%. Similarly for ZH-AT, the best accuracy is 97.17%. For MW-AA, the best accuracy recorded is 94.39% while for MW-AT, the best accuracy is 89.11%.

Table 5.11, shows results for all datasets for the threshold value of 13Å. For ZH-AA the best accuracy obtained is 95.38%. Similarly for ZH-AT, the best accuracy is 94.62%. For MW-AA, the best accuracy recorded is 90.07% while for MW-AT, the best accuracy is

Table 5.11: Classification results for MW and ZH datasets for a threshold value of 13Å.

		Linear Classifier LDR			Quadratic Classifier LDR		
Datasets	SVM	FDA	HDA	CDA	FDA	HDA	CDA
ZH-AA	95.38	88.46	88.46	93.08	86.15	89.23	92.31
ZH-AT	93.08	72.31	75.38	94.61	73.85	78.46	94.62
MW-AA	90.07	71.85	54.97	49.34	69.21	76.49	70.53
MW-AT	93.03	86.47	87.79	92.74	86.80	87.46	92.74

93.03%.

Table 5.12 shows the comparison for all threshold values ranging from 7Å to 13Å. The comparison is between two classification methods, SVM and LDR using different thresholds. For ZH-AA, the best accuracy is 96.18% for a threshold value of 9Å using SVM while for ZH-AT, the best accuracy obtained is 97.71% for a threshold value of 12Å using SVM. For MW-AA dataset, the best accuracy obtained is 99.67% for a threshold value of 9Å using SVM while for MW-AT, the best accuracy obtained is 97.36% for a threshold value of 10Å using SVM.

Table 5.12: Comparison of all threshold values from 7Å to 13Å.

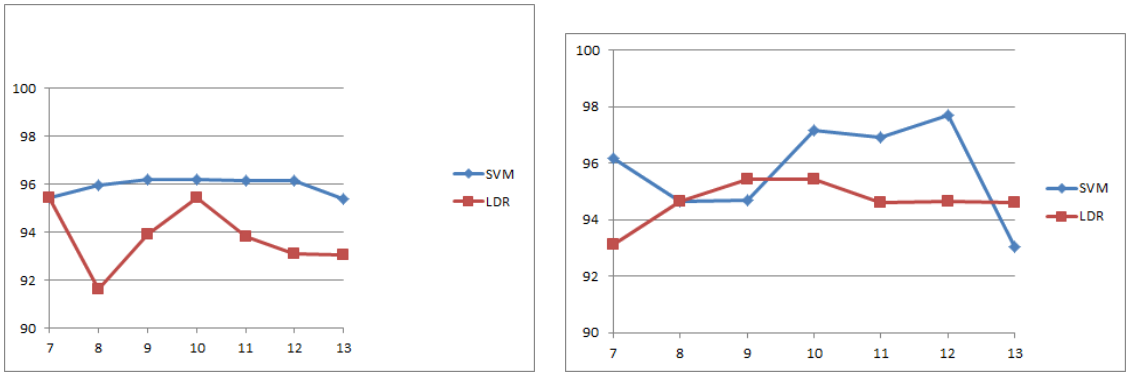
Datasets	Method	Inter-atomic distance thresholds						
		7Å	8Å	9Å	10Å	11Å	12Å	13Å
ZH-AA	SVM	95.42	95.95	96.18	96.17	96.12	96.15	95.38
	LDR	95.40	91.60	93.89	95.42	93.80	93.08	93.03
ZH-AT	SVM	96.18	94.66	94.68	97.17	96.92	97.71	93.03
	LDR	93.13	94.66	95.42	95.42	94.62	94.66	94.62
MW-AA	SVM	90.73	99.66	99.67	95.38	94.72	94.39	90.07
	LDR	83.11	98.02	98.02	94.72	84.16	87.79	76.49
MW-AT	SVM	95.03	93.03	92.74	97.36	94.06	89.11	93.03
	LDR	94.37	94.06	96.37	98.68	94.02	88.12	92.74

Figure 5.1 shows the classification accuracy plots for the ZH-AA and ZH-AT datasets. The *x*-axis shows the threshold values of distance calculated between atom pairs of interacting chains, which ranges from 7Å to 13Å. The *y*-axis shows the accuracy in a range

from 90% to 100%. Figure 5.1 (a) is for the ZH-AA dataset. Observing accuracies for SVM (shown in blue), it remains constant around 96% for most of the threshold values, but for LDR (shown in red), it varies from 88% to 95%. Figure 5.1 (b) represents the classification accuracy plots for the ZH-AT dataset. The accuracy plot for SVM (shown in red) achieves good accuracy of around 97% for the threshold value of 10Å and 12Å , while for LDR (shown in green), it lies below 96%. In a nutshell, SVM gives better accuracies for classification as compared to LDR for distinguishing between obligate and non-obligate interactions for ZH datasets.

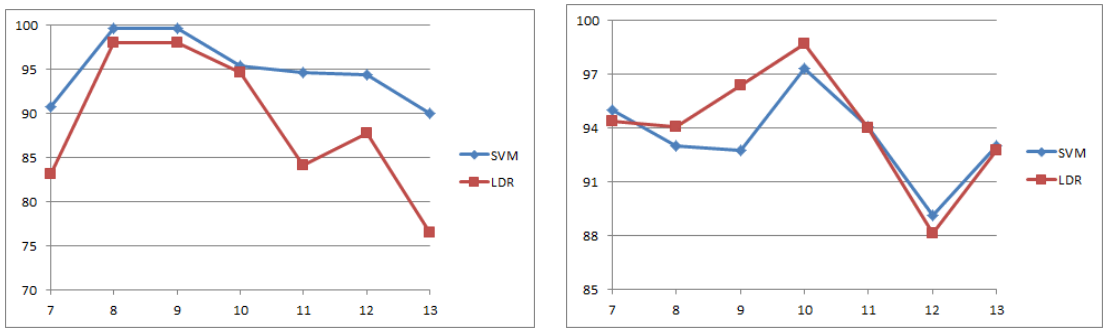
Figure 5.2 shows the classification accuracies plots for MW-AA and MW-AT datasets respectively. The *x*-axis shows the threshold values of distance calculated between atom pairs of interacting chains, which varies from 7Å to 13Å. The y-axis shows the accuracy range from 70% to 100% for Figure 5.2 (a) and from 85% to 100% for Figure 5.2 (b). Figure 5.2 (a) is for the MW-AA dataset. Observing the accuracies for SVM (shown in blue), for the threshold values of 8Å and 9Å it has the maximum accuracy of around 99%, but for LDR (shown in red), it shows huge variations from 76% to 98%. Figure 5.2 (b) represents the classification accuracies plots for the ZH-AT dataset. The classification accuracy plots for SVM (shown in blue) and LDR (shown in red) follows a similar pattern with slight variations.

The time required to run the entire PPIEE model for a single complex is around 8-12 minutes depending on the complex. Some complexes were tried on PPIEE model just to check the processing times.



(a) Accuracy plots for ZH-AA. (b) Accuracy plots for ZH-AT.

Figure 5.1: Classification accuracy plots for SVM and LDR on ZH-AA and ZH-AT datasets.



(a) Accuracy plot for MW-AA. (b) Accuracy plots for MW-AT.

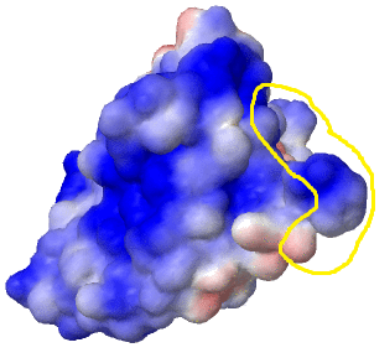
Figure 5.2: Classification accuracy plots for SVM and LDR on MW-AA and MW-AT datasets.

5.1.4 Visual Analysis

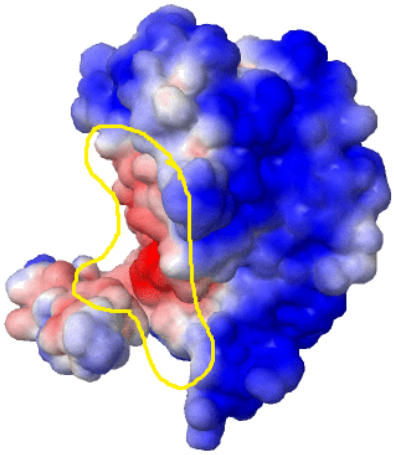
To study the effect of electrostatic energies for prediction from a different perspective, a visual analysis has been done. The analysis is done for an obligate complex PDB ID 2min and non-obligate complex PDB ID 1a2k from the MW dataset. Figure 5.3 shows the obligate complex along with the electrostatic potential for three different cases: Figure 5.3(a) shows one subunit (Chain A), Figure 5.3(b) represents another subunit (Chain B), and Fig-

ure 5.3(c) depicts Chains A and B combined. To visualize the effect of electrostatic energies for prediction, I show these proteins plotted over solvent accessible surface area, generated with the help of Jmol embedded in APBS. Observing Figure 5.3(a) carefully, the highlighted yellow portion has positive electrostatic potential (shown in blue), while in Figure 5.3(b), the highlighted yellow portion has negative electrostatic potential (shown in red). The interaction between the two chains takes place at these regions as shown in Figure 5.3(c). The positive and negative potentials on the corresponding areas of the interface of A and B yields very high affinity, and hence a favourable scenario for the obligate complex. This is the main feature that PPIEE exploits to predict the stability of protein complexes and it is corroborated in the experimental results.

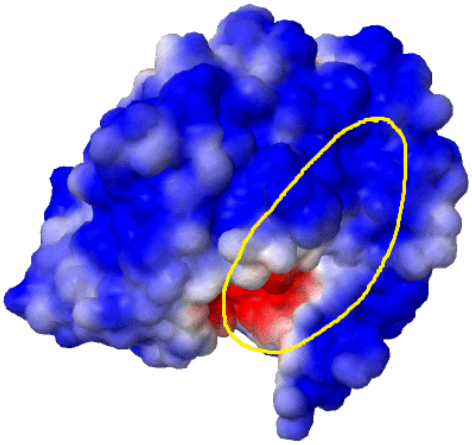
Figure 5.4 shows an non-obligate complex PDB ID 1a2k along with the electrostatic potential for three different cases: Figure 5.4(a) shows one subunit (Chain AB), Figure 5.4(b) represents another subunit (Chain C) and Figure 5.4 (c) depicts Chains AB and C combined. To visualize the effect of electrostatic energies for prediction, I show these protein complexes plotted over solvent accessible surface area, generated with the help of Jmol embedded in APBS. Observing Figure 5.4(a) carefully, the highlighted yellow portion has negative electrostatic potential (shown in red), while in Figure 5.4(b), the highlighted yellow portion has positive electrostatic potential (shown in blue). The interaction between the two chains takes place at these regions and shown in Figure 5.4(c).



(a) Electrostatic potential of 2min (chain A).

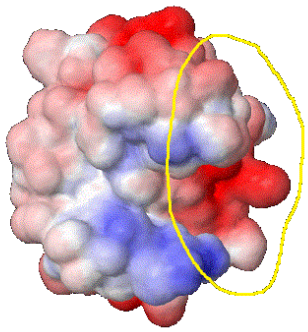


(b) Electrostatic potential of 2min (chain B).

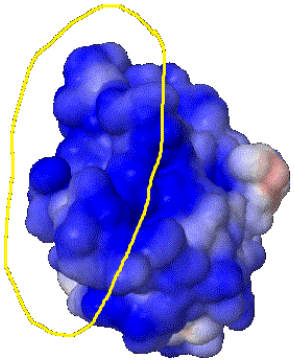


(c) Electrostatic potential of 2min (chains A and B).

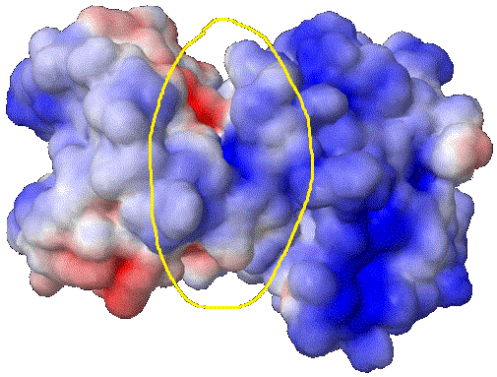
Figure 5.3: Electrostatic potential of an obligate complex, PDB ID 2min, plotted over solvent accessible surface area before and after the interaction takes place. The plots were generated by Jmol embedded in APBS.



(a) Electrostatic potential of 1a2k (chain AB).



(b) Electrostatic potential of 1a2k (chain C).



(c) Electrostatic potential of 1a2k (chains AB and C).

Figure 5.4: Electrostatic potential of a non-obligate complex, PDB ID 1a2k, plotted over solvent accessible surface area before and after the interaction takes place. The plots were generated by Jmol embedded in APBS.

5.1.5 Discussions

This section has some in depth discussions about the results which are obtained by varying the thresholds values and some visual analysis. In literature many other descriptors such as desolvation energies, hydrophobicity, solvent accessible surface area, amino acid composition and others have been reported which yields good classification accuracies for prediction of PPIs. Many researchers like Rueda *et al.* [43] and Liu *et al.* [26] used interface properties such as solvent accessible surface area to predict PPIs and obtained the best accuracy of around 81% and 92% respectively. PPIEE uses electrostatic energies as properties to predict types. Since the electrostatic energies are considered to be long ranged interactions, it might go upto 10Å and even more. Thus, the first run of PPIEE was for the threshold value of 10Å using SVM and LDR as classifiers achieving the accuracy of around 96%. Varying of thresholds is done to validate the PPIEE model and then comparing the results with others. The best classification accuracies for prediction of interaction types for both MW and ZH datasets are obtained at 9Å and 10Å as shown in Table 5.12. PPIEE approach is better than other approaches for the reason that by increasing the threshold values, it covers more wider interface area. PPIEE shows that by including more interface atoms and their corresponding electrostatic energies yields better classification accuracies for prediction of PPIs.

Some visual analysis has been done to study interaction types from a different perspective. The analysis is for interface area for a obligate complex PDB ID 2min and a non-obligate complex PDB ID 1a2k from MW datasets. For visualization, electrostatic potential has been plotted over solvent accessible surface area before and after the interaction takes place. The plots in Figure 5.3 and Figure 5.4 are generated by ICM browser. The positive and negative potentials at different subunits yields high affinity and plays important

role in the interaction.

Chapter 6

Conclusion and Future work

6.1 Summary of Contributions

The newly proposed PPIEE model works well for distinguishing protein interaction types. PPIEE uses electrostatic energies as properties for pairs of atoms and amino acids present in the interface. The classification is performed via SVM and LDR methods. Results from all the experiments suggest that, electrostatic energies turned out to be the best properties for prediction of protein interaction types.

The key contributions of the thesis can be summarized as follows:

- The proposed PPIEE model is used to predict protein interaction types (obligate and non-obligate) using electrostatic energies as properties. PPIEE works very well for distinguishing protein interaction types.
- The prediction approach uses electrostatic energies as properties for pairs of atoms or amino acids present in the interfaces of such complexes.

- The use of PDB2PQR [11] and APBS [6] to compute electrostatic energies for samples in the datasets.
- To integrate data from four different file formats named as PDB, PQR, OUT and INTERFACE file formats.
- Applied classification using SVM and LDR coupled with Bayesian to analyse results and predict the types.
- Changing the threshold distances from 7Å to 13Å and running the PPIEE model for each of them in order to have in-depth analysis for classification results.
- Visual analysis using electrostatic potential for some complexes in order to achieve in-depth analysis from a different perspective.

I observed that electrostatic energies turned out to be the best ones for prediction of interaction types on basis of all the experimental results. The reason why electrostatic energies give better prediction results is due to the fact that they are long ranged interactions which may go up to a 10 Å or even more. As a result, it covers a broader (and deeper) area in the interface yielding excellent results in classification. Also, they have more influence in polar and charged molecules. Thus, among various components of molecular interactions, electrostatic energies play a special role. The proposed features exploit the high affinity of proteins to interact with each other (in terms of negative and positive potentials).

6.2 Future Work

The future work involves various extensions to this thesis listed as follows:

- To study about domains and motifs present in the interface in order to achieve a better insight on proteins, their interactions and functions.
- Some post analysis on the available datasets to obtain a relevant pair of atom type and amino acid type that are biologically meaningful.
- Different feature selection methods such as sequential forward/backward search selection can be coupled with PPIEE to select the best subset of features that represents the whole feature set efficiently.
- Biologically guided feature selection and interpretation combined with automatic feature selection.
- Using electrostatic energies for the prediction of PPI of other types of interactions.

Appendix A

Explanation of the code

In this section, the explanation of the source code is given. The code works in various steps:

Step 1: DATASETS:

Initially, there is a list of complexes for ZH and MW datasets. From the list of complexes, I downloaded all the PDB files from PDB server. Then, I split those files based on their chain names. For example, if the PDB ID of the complex is 1a4y A:B, then, there will be two corresponding PDB files 1a4y:A and 1a4y:B. Now, using the script called Distance calculates, the atoms within a certain threshold are calculated based on the need. The INTERFACE file now contains the list of interface atoms as calculated by the script along with the exact distance.

Step 2: RUNNING THE SOFTWARE PACKAGES: PDB2PQR and APBS:

Now for these PDB files, I ran the scripts called PDB2PQR and APBS. The input to these servers are PDB files. After running both programs, I obtained PQR and OUT files respectively. The script called PDB2PQR will run the locally installed server of PDB2PQR and will output PQR files and IN files. Now these PQR files and IN files act as an input to

APBS which produces OUT files containing the electrostatic values. I ran separate scripts called APBS which runs the server APBS. I need to take care of the pre-processing the files at some steps, so that the software recognizes the files.

Step 3: a) CORRESPONDENCE SCRIPTS:

This involves four scripts, which go through all these four files and pick the exact electrostatic values for the interface atoms present in interface files. The pair of atoms that are present on the interface are listed in the INTERFACE file and their corresponding electrostatic values are present in the OUT file. Thus, in order to obtain the exact electrostatic values, this code goes through these four files to extract the exact electrostatic values for the pair of interface atoms. PPIEE uses Binary Search and also the extended version of Binary Search called SearchPQR that makes the search easier between these files. The names of the four scripts are Main.m , Main2.m, EVvalue1, EVvalue2, Bsearch.m, SearchPQR and Convertatom-type.m. Main.m and Main2.m are only used for reading the complexes from the file. The files called EVvalue.m and EVvalue2.m output electrostatic values and its corresponding types for atoms on interface. At the same time, I take care of saving the corresponding types for the atoms. I am considering two types, i.e. one is the atom type and the other is the amino acid type. Thus, for each interface atom, I have the corresponding atom-type and amino-acid type saved along with its electrostatic values. This information helps perform the classification. The script which output the corresponding atom type is called convertatom-type.m.

Step 4: FEATURE GENERATION:

Once I have the electrostatic values and its corresponding unique types, I need to create the feature vectors which act as input for the classifiers. Since I am working with a two

class problem, the two classes are obligate and non-obligate. For atom type, I have a matrix of 18 $\times$ 18 dimensions since there are 18 unique types of atoms. Then for each pair of atoms, I average the electrostatic values and put in the matrix. For repeated pairs of atoms, I keep accumulating the values at the same position in the matrix. The scripts that perform these jobs are CheckAdd.m, checkexistingposition.m, findposition1.m and makefeatures.m. These scripts will ultimately produce the features for each complex. Once all the features are created, the script combinefeatures.m combines all the features that produce the dataset. I do the same job of the amino acid type and make a matrix for it in the same way. In this case, I have a matrix of 20 $\times$ 20, since there are 20 standard amino acids.

Step 5: CLASSIFICATION:

When the matrices are ready for both atom type and amino acid type, I am in a good state to apply classification on these matrices. Since I know before hand which class the particular complex belongs to, I can attach labels to the features. Labels are important since they help in classification. I provide the matrices as input to the script MainSVM.m. This script performs classification on the the matrix and outputs the accuracy for the classifier for predicting the interaction types. In the experiments, I use two classifiers, SVM and LDR. In order to run the SVM, I run the Function MainSVM.m. This function will use different kernels to apply classification. It uses linear, polynomial and radial basis function kernels and outputs the results. For the LDR, I run the function LDR.m which use different criteria such as FDA, HDA and CDA and outputs the classification accuracy is in a text file. The only aspect I need to make sure is that labels are attached to the datasets. All these functions are automated and I only need to provide the name of the matrix and attach the labels to it.

The source code and datasets used for PPIEE are available for downloading at: `http`:

//cs.uwindsor.ca/ ~lrueda/software/PPIEE.zip

Bibliography

[1] Affinity Chromatography Principles and Methods. Amersham Pharmacia Biotech, 2010.

[2] O. K. A. Zen, C. Micheletti and R. Nussinov. Comparing interfacial dynamics in protein-protein complexes: an elastic network approach. BMC Structural Biology, 10 (26), 2010. doi: 10.1186/1472-6807-10-26.

[3] S. Abe. Support Vector Machines for Pattern Classification (Advances in Pattern Recognition). Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 1852339292.

[4] M. M. Aziz, M. Maleki, L. Rueda, M. Raza, and S. Banerjee. Prediction of biological protein-protein interactions using atom-type and amino acid properties. Proteomics 2011, 11:17–22, 2011.

[5] N. Baker. Continuum models for biomolecular solvation. 2008. URL http://www.scitopics.com/Continuum_models_for_biomolecular_solvation.html . Pacific Northwest National Laboratory.

[6] N. A. Baker, D. Sept, S. Joseph, M. J. Holst, and J. A. McCammon. Electrostatics of

nanosystems: Application to microtubules and the ribosome. 98(18):10037–10041, 2001. doi: 10.1073/pnas.181342398.

[7] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne. The Protein Data Bank. Nucleic Acids Research, 28:235–242, 2000.

[8] C. Camacho and C. Zhang. FastContact: rapid estimate of contact and binding free energies. Bioinformatics, 21(10):2534–2536, 2005.

[9] N. E. Davey, K. Van Roey, R. J. Weatheritt, G. Toedt, B. Uyar, B. Altenberg, A. Budd, F. Diella, H. Dinkel, and T. J. Gibson. Attributes of short linear motifs. Mol. BioSyst., 8:268–281, 2012. doi: 10.1039/C1MB05231D.

[10] F. P. Davis and A. Sali. Pibase: a comprehensive database of structurally defined protein interfaces. 21(9):1901–1907. doi: 10.1093/bioinformatics/bti277.

[11] T. J. Dolinsky, P. Czodrowski, H. Li, J. E. Nielsen, J. H. Jensen, G. Klebe, and N. A. Baker. Pdb2pqr: expanding and upgrading automated preparation of biomolecular structures for molecular simulations. 35(suppl 2):W522–W525, 2007. doi: 10.1093/nar/gkm276.

[12] R. Duda, P. Hart, and D. Stork. Pattern Classification. John Wiley and Sons, Inc., New York, NY, 2nd edition, 2000.

[13] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification (2nd Edition). Wiley-Interscience, 2 edition, Nov. . ISBN 0471056693.

[14] V. Estivill-Castro. Why so many clustering algorithms: a position paper. SIGKDD Explor. Newsl., 4(1):65–75, June 2002. ISSN 1931-0145.

[15] R. Fisher. The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics, 7:179–188, 1936.

[16] G. Geva and R. Sharan. Identifcation of protein complexes from co-immunoprecipitation data. BMC Systems Biology, 27(1):111–117, 2010.

[17] E. Golemis. Protein-protein intercations: a molecular cloning manual. Cold Spring Harbour Laboratory Press, second edition, 2005.

[18] M. N. Isabelle Guyon, Steve Gunn and L. Zadeh. Feature Extraction, Foundations and Applications. Springer, first edition, 2006.

[19] O. Ivanciuc. Applications of support vector machines in chemistry. Reviews in computational chemistry, pages 291–400, 2007.

[20] S. P. B. A. G. M. L. James D. Watson, Tania A. Baker and R. Losick. Molecular Biology of the Gene. Benjamin Cummings, sixth edition, 2008.

[21] I. Jolliffe. Principal Component Analysis. Springer, second edition, 2002.

[22] S. Jones and J. M. Thornton. Principles of protein-protein interactions. Proc. Natl Acad. Sci, USA, 93(1):13–20, 1996.

[23] A. Kessel and N. Ben-Tal. Introduction to Proteins: Structure, Function, and Motion. CRC Press, 2010.

[24] I. Kurareva and R. Abagyan. Predicting molecular interactions in structural proteomics. In R. Nussinov and G. Shreiber, editors, Computational Protein-Protein Interactions, chapter 10, pages 185–209. CRC Press, 2009.

[25] M. C. Lawrence and P. M. Colman. Shape complementarity at protein/protein interfaces. J. Mol Biol, 234(4):946–950, 1993.

[26] Q. Liu and J. Li. Propensity vectors of low-asa residue pairs in the distinction of protein interactions. Proteins: Structure, Function, and Bioinformatics, 78(3). ISSN 1097-0134.

[27] C. D. Livingstone and G. J. Barton. Protein sequence alignments: a strategy for the hierarchical analysis of residue conservation. 9(6):745–756, 1993. doi: 10.1093/bioinformatics/9.6.745.

[28] M. Loog and P. Duin. Linear Dimensionality Reduction via a Heteroscedastic Extension of LDA: The Chernoff Criterion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(6):732–739, 2004.

[29] M. Maleki and L. Rueda. Domain-domain interactions in obligate and non-obligate protein-protein interactions. In Bioinformatics and Biomedicine Workshops (BIBMW), 2011 IEEE International Conference on, pages 907 –908, nov. 2011. doi: 10.1109/BIBMW.2011.6112498.

[30] M. Maleki, M. Aziz, and L. Rueda. Analysis of relevant physicochemical properties in obligate and non-obligate protein-protein interactions. In Bioinformatics and Biomedicine Workshops (BIBMW), 2011 IEEE International Conference on, pages 345 –351, nov. 2011. doi: 10.1109/BIBMW.2011.6112397.

[31] J. Mintseris and Z. Weng. Structure, function, and evolution of transient and obligate protein-protein interactions. Proc Natl Acad Sci, USA, 102(31):10930–10935, 2005.

[32] J. Mintseris and Z. Weng. Structure, Function, and Evolution of Transient and Obligate Protein-protein Interactions. Proceedings of the National Academy of Sciences, USA, 102(31):10930–10935, 2005.

[33] I. Nooren and J. Thornton. Diversity of protein-protein interactions. EMBO Journal, 22(14):3846–3892, 2003.

[34] R. Nussinov and G. Schreiber. Computational Protein-Protein Interactions. CRC Press, 2009.

[35] Y. Ofran. Prediction of protein interaction sites. In R. Nussinov and G. Shreiber, editors, Computational Protein-Protein Interactions, chapter 9, pages 167–184. CRC Press, 2009.

[36] Y. Ofran and B. Rost. Analysing six types of protein-protein interfaces. Journal of molecular biology, 325(2):377–387, 2003.

[37] S. Park, J. Reyes, D. Gilbert, J. Kim, and S. Kim. Prediction of protein-protein interaction types using association rule based classification. BMC Bioinformatics, 10(1): 36, 2009.

[38] G. A. Petsko and D. Ringe. Protein structure and function. New Science Press, 2004.

[39] M. Punta, P. C. Coggill, R. Y. Eberhardt, J. Mistry, J. Tate, C. Boursnell, N. Pang, K. Forslund, G. Ceric, J. Clements, A. Heger, L. Holm, E. L. L. Sonnhammer, S. R. Eddy, A. Bateman, and R. D. Finn. The pfam protein families database. 40(D1): D290–D301, 2012. doi: 10.1093/nar/gkr1065.

[40] L. A. H. R. Benjamin Free and D. R. Sibley. Identifying Novel Protein-Protein Interactions Using Co-Immunoprecipitation and Mass Spectroscopy. Current Protocols in Neuroscience, page DOI: 10.1002/0471142301.ns0528s4, 2009.

[41] L. Rueda and M. Herrera. Linear Dimensionality Reduction by Maximizing the Chernoff Distance in the Transformed Space. Pattern Recognition, 41(10):3138–3152, 2008.

[42] L. Rueda, S. Banerjee, M. M. Aziz, and M. Raza. Protein-protein interaction prediction using desolvation energies and interface properties. Proceedings of the 2nd. IEEE International Conference on Bioinformatics & Biomedicine (BIBM 2010), pp. 17–22, 2010.

[43] L. Rueda, S. Banerjee, M. Aziz, and M. Raza. Protein-protein interaction prediction using desolvation energies and interface properties. In Bioinformatics and Biomedicine (BIBM), 2010 IEEE International Conference on, pages 17 –22, dec. 2010. doi: 10.1109/BIBM.2010.5706528.

[44] L. Rueda, C. Garate, Banerjee, and M. M. Aziz. Biological protein-protein interaction prediction using binding free energies and linear dimensionality reduction. Proceedings of the 5th. IAPR International Conference on Pattern Recognition in Bioinformatics (PRIB 2010), pages 383–394, 2010.

[45] K. K. Sergios Theodoridis. Pattern Recognition. Academic press,Inc. Orlando, Fl, USA, third edition, 2006.

[46] E. Tropp and D. Freifelder. Molecular Biology: Genes to Proteins. Jones and Bartlett Learning, fourth edition, 2008.

[47] L. Wong. The Practical Bioinformatician. World Scientific Publishing Co. Pte. Ltd., first edition, 2004.

[48] S. Xu, T. Jin, and F. C. M. Lau. A new visual search interface for web browsing. In Proceedings of the Second ACM International Conference on Web Search and Data Mining, WSDM '09, pages 152–161, 2009. ISBN 978-1-60558-390-7.

[49] C. Zhang, G. Vasmatzis, J. L.Cornette, and C. DeLisi. Determination of atomic desolvation energies from the structures of crystallized proteins. J. Mol. Biol., 267: 707–726, 1997.

[50] H. Zhu, F. Domingues, I. Sommer, and T. Lengauer. Noxclass: Prediction of protein-protein interaction types. BMC Bioinformatics, 7(27), 2006. doi:10.1186/1471-2105-7-27.

Vita Auctoris

Gokul Vasudev was born in 1986 in Panchkula, India. He received his Bachelors degree from Punjab Technical University in Computer Science and Engineering from in 2009. His research interests include pattern recognition, protein-protein interaction, machine learning and bioinformatics.