Machine Learning Methods for Prediction of Protein-Protein Interaction Hot Spot Residues.

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades einer Doktorin der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Divya Sitani, M.Tech.

aus

Allahabad, Indien

Berichter:
Prof. Dr. Geraldine Zimmer-Bensch
Prof. Dr. Paolo Carloni

Tag der mündlichen Prüfung: 16.01.2024

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek verfügbar.

Acknowledgments

First and foremost, I would like to thank my supervisor Prof. Dr. Paolo Carloni for his constant support and encouragement. He demonstrated complete faith in my ideas and guided me in successfully implementing them always.
I would like to express my sincere gratitude to Prof. Dr. Geraldine Zimmer-Bensch from the department of Biology at RWTH Aachen for supervising me and giving me meaningful insights about the thesis that helped me significantly to improve this thesis.
I would like to thank my co-supervisor, Dr. Mercedes Alfonso-Prieto. Her help at various stages during this work, right from the first day of my PhD, has been extremely valuable. She was always ready to sit through even extremely long meetings to clear all my queries.
I am really grateful to my closest friends, Mrs. Petra Rott, Dr. Somayeh Ashgharpour and Dr. Varun Loomba, who have always looked out for me and helped me professionally and emotionally.
I am grateful to my colleagues in the Computational Biomedicine group, INM-9/IAS-5 at Forschungszentrum Juelich, for their substantial scientific discussions and support throughout various stages of my PhD. I would like to express my appreciation to Dr. Emiliano Ippoliti, Mr. Markus Thoma, Dr. Jakob Schneider, and Dr. Volker Backes for always offering their technical and administrative assistance whenever needed.
This acknowledgement would be incomplete without expressing my immense gratitude to my husband, Dr. Protim Bhattacharjee for having invigorating and long scientific discussions with me and for not only believing in me but also making me believe in my abilities and strengths always. Thank you for always showing me light even at the end of the darkest of tunnels. I truly have no idea what I would have done without you.
I dedicate this thesis to my mother, Mrs. Jyoti Rani Sitani, who has constantly been an unending source of inspiration and love for me. I am deeply indebted to you, Ma, for your unwavering belief in my abilities and for providing me with the strength to believe in myself and pursue my dreams.
This thesis could not have been completed without my sister, Dr. Keerti Sitani, whose passion for science and her consistent dedication to her research have been a constant source of inspiration for me. Her unwavering commitment to her work motivates me to strive for self-improvement and work harder every day.
I wish to express my gratitude to my father, Mr. Puran Chand Sitani, who, despite not having the opportunity to pursue higher education himself due to family responsibilities, unfailingly inquired about the progress of my thesis and research every time we spoke. Thank you for always motivating me to stay focussed on my work and standing beside me through all the ups and downs of life.
Last but not least, I would like to extend my gratitude to all my professors, teachers, seniors, juniors, friends, and family who have supported me and aided in my personal growth throughout my life.

Abstract

Protein–protein interactions (PPIs) form a vast and intricate network of reactions important for the regulation and execution of most biological processes [Rao+14]. PPIs occur when two proteins make direct physical contact via their surface residues and form an interface, which is a non-uniform surface on a protein-protein complex [GS10]. Even though a protein interface may occupy a large area, only a small subset of its buried residues plays a crucial role in the binding free energy of the complex [BT98; Jan95]. These energetically key residues are known as hot spots. The experimental method to identify them is Alanine Scanning Mutagenesis (ASM) where systematically each interface residue is mutated to Alanine and the consequent change in binding energy $\Delta\Delta G_{binding}$ between the wild type and the mutant complex is measured. If ($\Delta\Delta G_{binding}$) is larger than a certain threshold, typically 2 kcal/mol, the interface residue is defined as a hot spot or else it is considered a null spot [MFR07; CW89; BT98]. The so-called hot spot residues are often enriched in disease-associated mutations [Ten+09]. These mutations often cause disrupted or erroneous protein interactions, resulting in phenotypic changes that might cause a disease. Moreover, with the discovery of hot spots in protein-protein interfaces, it has become possible to target a broader range of PPIs with small molecule drugs. The identification of hot spots has helped researchers to identify molecules that interact at these sites, thus interfering with PPIs and the downstream pathways they mediate [Pet+16a; Pet+16b; Sco+16]. Therefore, predicting hot spots is crucial to understand the effect of disease-associated mutations on PPIs and for drug discovery [Mur+17]. As mentioned before, experimentally hot spots can be found out by using ASM, but it is quite costly and tedious and this has led to the use of computational methods to predict hot spot residues. Previous computational approaches included molecular dynamics and knowledge based methods [GNS02; KB02; MK99; HMK02; GF08; Bre+09]. However, such approaches were time-consuming and hence limited in the number of hot spots predicted. This led to an increased use of machine learning (ML) based methods for hot spot prediction in recent years [DPM07; Den+13; CKL09a; CKL09b; Ass+10]. Such ML approaches capitalize on the availability of experimental datasets containing protein-protein complex structures and ASM-derived hotspot data. However, as it often happens with biological data repositories, such hotspot datasets often contain noise [Mor+17; KC21]. If machine learning (ML) algorithms are trained and predictions are made on this "noisy" data, the results will not be accurate [GG19]. The earlier ML based approaches for hot spot prediction did not take this issue into account. In this thesis, I describe the basic concepts and recent advances of machine learning applications in finding the protein–protein interaction hot spots. To reduce the effects of noise in hot spot prediction, I have proposed the method RBHS (**R**obust Principal Component Analysis-(RPCA) **b**ased Prediction of Protein-Protein Interaction **H**ot **S**pots)) in this thesis [Sit+21]. I use RPCA [Can+11] followed by feature selection using Extreme Gradient Boosting (XGBoost) [CG16] on the data matrix containing protein sequence and structure based features calculated on the interface residues. I trained several popular machine learning classifiers on the

benchmark dataset HB-34 [LLD18] and evaluated the performance of my proposed method on the independent test set BID-18 [LLD18]. After extensive computational experimentation and comparison with the existing state-of-the-art approaches to predict hot spots, I was able to show that my method is quite efficient in identifying hot spot residues crucial for protein-protein interactions. Finally, I discuss the challenges and future directions in the prediction of hot spots in this thesis.

Zusammenfassung

Protein-Protein-Interaktionen (PPI) bilden ein umfangreiches und kompliziertes Netz von Reaktionen, die für die Regulierung und Ausführung der meisten biologischen Prozesse wichtig sind [Rao+14]. PPIs treten auf, wenn zwei Proteine über ihre Oberflächenreste in direkten physischen Kontakt treten und eine Grenzfläche bilden, d. h. eine ungleichmäßige Oberfläche auf einem Protein-Protein-Komplex [GS10]. Obwohl eine Proteinschnittstelle eine große Fläche einnehmen kann, spielt nur eine kleine Untergruppe der darin enthaltenen Reste eine entscheidende Rolle für die freie Enthalpie der Bindung des Komplexes [BT98; Jan95]. Diese Reste werden als Hot Spots bezeichnet. Die experimentelle Methode zu ihrer Identifizierung ist die Alanin-Scanning-Mutagenese (ASM), bei der systematisch jeder schnittstellenrest zu Alanin mutiert und die daraus resultierende Änderung der Bindungsenergie $\Delta\Delta G_{binding}$ zwischen dem Wildtyp und dem mutierten Komplex gemessen wird. Ist ($\Delta\Delta G_{binding}$) größer als ein bestimmter Schwellenwert, in der Regel 2 kcal/mol, wird der Schnittstellen rest als Hot-Spot definiert, andernfalls wird er als Null-Spot [MFR07; CW89; BT98] betrachtet. Die so genannten Hot-Spot-Reste sind häufig mit krankheitsassoziierten Mutationen beteiligt [Ten+09]. Diese Mutationen führen oft zu gestörten oder fehlerhaften Proteininteraktionen, was zu phänotypischen Veränderungen führt, die eine Krankheit verursachen können. Mit der Entdeckung von Hot Spots in Protein-Protein-Schnittstellen ist es außerdem möglich geworden, eine breitere Palette von PPIs mit kleinen Molekülen zu beeinflussen. Die Identifizierung von Hot Spots hat den Forschern geholfen, Moleküle zu identifizieren, die an diesen Stellen interagieren und so die PPIs und die nachgeschalteten Stoffwechselwege stören [Pet+16a; Pet+16b; Sco+16]. Daher ist die Vorhersage von Hot Spots von entscheidender Bedeutung für das Verständnis der Auswirkungen von krankheitsassoziierten Mutationen auf PPIs und für die Entwicklung von Medikamenten [Mur+17]. Wie bereits erwähnt, können Hot Spots experimentell mit Hilfe von ASM ermittelt werden. Dies ist jedoch recht kostspielig und langwierig, was zum Einsatz von Berechnungsmethoden zur Vorhersage von Hot Spot-Resten führte. Zu den früheren Berechnungsmethoden gehörten Molekulardynamik und wissensbasierte Methoden [MK99; HMK02; GF08; Bre+09]. Solche Ansätze waren jedoch zeitaufwändig und daher in der Anzahl der vorhergesagten Hot Spots begrenzt. Dies führte dazu, dass in den letzten Jahren verstärkt Methoden des maschinellen Lernens (ML) für die Hot-Spot-Vorhersage eingesetzt wurden [DPM07; Den+13; CKL09a; CKL09b; Ass+10]. Solche ML-Ansätze machen sich die Verfügbarkeit experimenteller Datensätze zunutze, die Protein-Protein-Komplexstrukturen und von ASM abgeleitete Hotspot-Daten enthalten. Wie bei biologischen Datenbeständen üblich, enthalten solche Hotspot-Datensätze jedoch häufig Rauschen [Mor+17; KC21]. Wenn Algorithmen des maschinellen Lernens (ML) auf diesen "verrauschten" Daten trainiert werden und Vorhersagen getroffen werden, sind die Ergebnisse nicht genau [GG19]. Bei den früheren ML-basierten Ansätzen zur Vorhersage von Hotspots wurde dieses Problem nicht berücksichtigt. In dieser Arbeit beschreibe ich die grundlegenden Konzepte und die jüngsten Fortschritte der Anwendungen des maschinellen Lernens bei der Suche nach den Protein-Protein-

Interaktions-Hotspots. Um die Auswirkungen des Rauschens bei der Vorhersage von Hot Spots zu reduzieren, habe ich in dieser Arbeit die Methode RBHS (**R**obust Principal Component Analysis-(RPCA) **b**ased Prediction of Protein-Protein Interaction **H**ot **S**pots) vorgeschlagen [Sit+21]. Ich wende RPCA [Can+11], gefolgt von einer Merkmalsauswahl mit Extreme Gradient Boosting (XGBoost) [CG16] auf die Datenmatrix an, die Proteinsequenz- und strukturbasierte Merkmale enthält, die für die Schnittstellenreste berechnet wurden. Ich habe mehrere gängige Klassifikatoren für maschinelles Lernen auf dem Benchmark-Datensatz HB-34 [LLD18] trainiert und die Leistung der von mir vorgeschlagenen Methode auf dem unabhängigen Testsatz BID-18 [LLD18] bewertet. Nach ausgiebigen Experimenten und einem Vergleich mit den bestehenden State-of-the-Art-Ansätzen zur Vorhersage von Hot Spots konnte ich zeigen, dass meine Methode bei der Identifizierung von Hot Spot-Resten, die für Protein-Protein-Interaktionen entscheidend sind, recht effizient ist. Abschließend diskutiere ich in dieser Arbeit die Herausforderungen und zukünftigen Richtungen bei der Vorhersage von Hot Spots.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Hot Spots and their relevance

Rarely do proteins operate alone [Les10]. To perform highly diverse biological functions, from metabolism and signal transduction to cellular motility and synaptic transmission along with other cell-cell interactions, proteins frequently interact with other proteins and biomolecules [Sti97; Jan95]. This leads to an intricate network of interactions, and the complete set of protein-protein interactions in a living organism is known as the interactome. Thus, protein-protein interactions are very essential for performing various biological processes including cell to cell interactions, metabolic and developmental control [Rao+14].

Proteins-protein interactions (PPIs) take place on these particular regions of the protein surface known as protein interfaces [GS10; DS15]. Interfaces are characterized by three regions known as core, rim and support. The interface core and rim differ not only in their energetic contribution to the complex stability, but also in terms of their physicochemical and evolutionary characteristics [DS15]. Interface core residues become solvent inaccessible upon protein-protein interaction, whereas interface rim residues remain partially solvent accessible [DS15; MFR07]. Interface core residues tend to be evolutionarily more conserved and their side chains often display less mobility upon binding, compared to rim residues. The support region is identified in [Lev10] is formed by partially exposed residues in the unbound protein that become buried when the complex is formed, and thus can facilitate protein-protein interactions. Even though protein interfaces may be large, it turns out that few residues do contribute significantly to the binding free energy of the complex ($\Delta\Delta G_{binding}$) [BT98; Jan95]. These key energetic residues are known as hot spots.

These structural details of interfaces are important when studying the role of disease associated mutation on protein-protein interactions because protein interfaces are often enriched in disease-causing mutations compared to other protein surface regions [DS15]. Mutations can disrupt a protein interface by modifying its physicochemical, structural, and energetic characteristics [DS15]. Moreover, disease-causing mutations are expected to have a greater impact on protein structure, function, and protein complex thermodynamics when occurring in hot spots [Ten+09].

Owing to the peculiar large and flat nature of protein interaction surfaces which are often missing features such as pockets, grooves, or clefts that could act as potential docking sites for small molecule inhibitors, protein-protein complexes could not be used as drug targets for a long time. Even if the features are present, the structural complexity of the interface poses challenges for modeling a new therapeutic molecule.

The lack of natural small ligands, which could be an alternative starting point of drug design, is another major obstacle [Pet+16a; WM07]. The discovery of hot spots in PPI interfaces, made it possible to target a broader range of PPIs with small molecule drugs. The identification of hot spots has enabled researchers to identify molecules that interact at these sites, thus interfering with PPIs and the downstream pathways they mediate [Pet+16a; Pet+16b; Sco+16]. Thus, predicting hot spots is crucial to understand the effect of disease-associated mutations on PPIs and drug discovery [Mur+17].

An example of this in cancer research is the protein complex between human murine double minute 2 (MDM2) and human p53. p53 is a transcription factor that plays a key role in cell cycle regulation, apoptosis, DNA repair, senescence, angiogenesis, and innate immunity [VL02; Bro+09]. p53 is a potent tumor suppressor but in a lot of human cancers, its antitumor activity is impaired due to the mutations within the p53 gene [FI04]. In other human cancers, it does retain its wild-type status, but its function of tumor suppression is compromised by a multitude of intracellular mechanisms.

MDM2 in humans is the major inhibitor of p53. It binds to p53 directly that results in a repressed p53 transactivation activity, enhanced nuclear export of p53, and degradation of p53 [Wu+93; FWL99; JO99]. Over expression of MDM2 in human tumors indicates poor clinical prognosis or poor treatment response to existing cancer treatments. Indeed, MDM2 is overexpressed in several cancer types [Mom+98]. Interfering with MDM2/p53 complex formation might improve the antitumor potency of p53.

MDM2 and p53 interact via a hydrophobic surface groove in MDM2 and three key hydrophobic residues in p53, Phe19, Trp23, and Leu26 [Cap+98; MWD00]. These residues constitute the hot spots that the researchers targeted in an effort to find molecules that can interrupt this particular interaction [Kus+96]. Though active research is still in progress, several MDM2-p53 inhibitors have moved onto clinical trials, showing decent results [Vu+13; Din+13; Zha+15; Sun+14; Wan+14]. Thus, the PPIs, if druggable can be inhibited or stabilized by targeting the hotspot residues.

1.2 Methods for Hot Spot Prediction

In the past, hot spot prediction was done using experimental methods. Due to limitations of experimental setups, quite a few computational approaches to predict hot spots became extremely popular. These include knowledge-based methods, molecular simulation techniques, and machine learning methods. Detailed information about these methods is provided in the next sections of this chapter.

1.2.1 Experimental Methods

The experimental method to identify hot spot residues is called Alanine Scanning Mutagenesis (ASM). Experimental ASM involves the systemic point mutation of protein-

protein interface residues to Alanine, which is then followed by expression and purification of mutants and measuring the change in $\Delta G_{binding}$ ($\Delta\Delta G_{binding}$=ΔG_{mutant}–$\Delta G_{wild\text{-}type}$) [BT98]. If ($\Delta\Delta G_{binding}$) ≥ 2.0 kcal/mol, the interface residue is defined as a hot spot, otherwise as a "null spot" [MFR07; CW89; BT98]. The disadvantage with ASM experiments, however, is that they are time-consuming and labor-intensive. Moreover, they are highly dependent on the used assays.

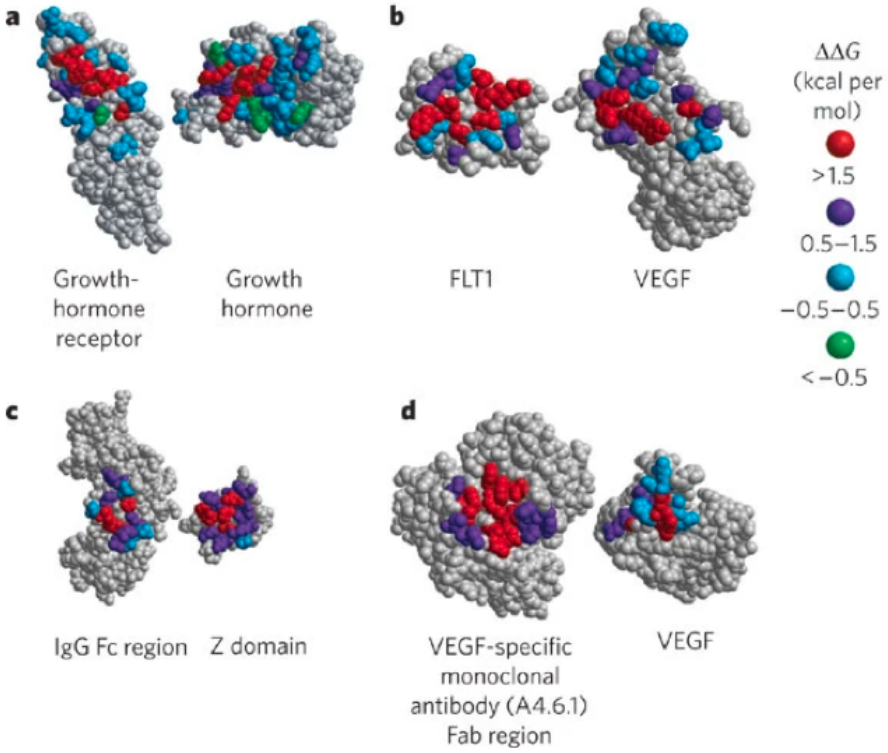


Figure 1.2.1: Examples of protein-protein interface hot spots. Alanine Scanning Mutagenesis was carried out on the contact surfaces of four pairs of interacting proteins. The resulting change in binding free energy $\Delta\Delta G_{binding}$ is shown by colour coding of interfacial amino acid residues. These colours range from red (indicating the most disruptive changes) to green (having little or no change). It can be seen from the figure that in each case only a small set of residues make a major contribution to binding free energy, i.e., the residues in red and these are the so-called hot spots. VEGF, Vascular Endothelial Growth Factor; Z domain, a derivative of a domain from Staphylococcus aureus protein A [WM07].

1.2.2 Computational methods

To overcome the problems pertaining to experimental procedures, in the last two decades there has been a significant rise in the use of *in silico* methods for hot spot

prediction. Some of these methods are knowledge-based computational methods, or they employ molecular dynamic simulations [GNS02; KB02; MK99; HMK02; GF08; Bre+09].

1.2.2.1 Molecular Dynamics based hot spot prediction

Molecular dynamics assumes that the motion of proteins can be simulated using classical physics. Molecular dynamics simulations can offer a detailed analysis of protein interfaces at the atomic level and estimate the changes in binding free energy ($\Delta\Delta G_{binding}$). Although molecular simulation methods [MK99; HMK02; GF08; Bre+09] might provide good predictive results, they are seldom applicable for large-scale hot spot predictions due to their huge computational cost.

1.2.2.2 Knowledge-based methods

Knowledge-based approaches like FOLDEF [GNS02] and Robetta [KB02] predict hot spots based on an estimate of the energetic contribution to binding for every interface residue.

FOLDEF [GNS02] was built on the FoldX complex energy function. Similarly, Robetta [KB02] uses a simple physical model. For both the methods, the predicted changes in binding energies after mutating side chains to Alanine form the basis for hot spot predictions [MZ12]. They do provide an alternative approach to predicting hot spots with less computational cost than molecular dynamics approaches. However, these are often still time-consuming and thus difficult to apply in high-throughput mode.

1.2.2.3 Machine Learning based hot spot prediction

The limitations in these previous computational methods have led to a substantial increase in the use of machine learning (ML) methods for the *in silico* prediction of hot spot residues. Moreover, since machine learning techniques can use data to learn without being externally programmed and there has been an increase in availability of data and more powerful hardware and software resources, this has led to an increased application of Machine Learning (ML) algorithms in every field including the task of predicting protein-protein interaction hot spot residues.

The first step of using ML for hot spot identification is to encode the various sequence and structure based features on the protein-protein interaction interface residues by the large number of available bio-informatics tools. Thereafter, these features are forwarded to a machine learning algorithm that learns to map such features to hot-spots.

1.3 State-of-the-art Machine Learning techniques for hot spot prediction

Here, I would like to discuss several popular hot spot prediction methods based on machine learning. To quantify the predictive power of my method, it is compared with these methods in Section 5.3.

1. KFC (Knowledge-based FADE and Contacts) [DPM07] uses a rule-based model called the KFC model that is a combination of two learned decision tree models. A decision tree [Has+09] (Section 1) is a predictive model that defines a set of Boolean tests or decisions to be taken at each step. The result of each test determines the next test to apply. The process continues until the path terminates, where the model predicts the class label for a sample. The two decision tree models in KFC are called K-FADE and K-CON trained on protein structure based features. K-FADE uses the features, residue size, and radial distribution of shape specificity and interface points calculated by Fast Atomic Density Evaluation (FADE) [MKT01], and K-CON uses the residue's intermolecular atomic contacts, hydrogen bonds, interface points, and chemical types.

2. The results of KFC were later on improved with two new models trained using a Support Vector Machine (SVM) [Guy+02; BGV92]. SVM is a widely used machine learning method that establishes the optimal hyperplane in a high-dimensional feature space to separate the data into two classes (hot spots and null spots). A detailed description of SVM along with its mathematical formulation is provided in Section 3.2.4.1. The two SVM models are KFC2a and KFC2b [ZM11]. KFC2a comprises eight features that are primarily related to solvent accessible surface area and local plasticity. KFC2b uses seven such features (two of which are common with KFC2a).

3. The authors of [CKL09a] applied an SVM that initially incorporates 54 protein structure and sequence based features. The method is called MINERVA (MINE Residue VAlue). Out of these 54 features, feature selection [SIL07; GE03] is done using a decision tree model to identify the best feature subset. Feature selection will be explained in Section 2 of Chapter 3.

4. Pred HS-SVM [Den+13] is another method that uses SVM, and here energy terms are used as input features of the SVM classifier. Their method considers basic energy terms such as van der Waals, H-bond, electrostatic and desolvation potentials, hydrogen bonds, and Coulomb electrostatics calculated from the protein complex's structure. The authors in [Xia+16] used 108 features based on protein sequences and structure based information and selected two highest-ranking features using a two-step feature selection method. The final prediction model was constructed by using the support vector machine and was called HEP.

5. In APIS (A combined model based on Protrusion Index and Solvent accessibility) [CKL09b] carefully studied 62 protein sequence and structure based features and then used the F1-Score to remove redundant features. F1-Score as defined in Section 3.2.5 is a performance metric used to quantify the predictive performance of a machine learning classifier. The APIS predictor uses an SVM classifier to identify hot spots.

6. Another method called PCRPi (Presaging Critical Residues in Protein interfaces) [Ass+10] is based on the integration of three main sources of information, namely, energetic, structural, and evolutionary determinants by using Bayesian networks to combine them into a common probabilistic framework. PCRPi can handle some of the missing protein data. This method has been developed as a web server (called PCRPi-W) [SAF10] where users can enter a PDB code or upload a complex, as well as select the type of Bayesian network architecture (naïve or expert) [Pea88].

1.4 Developed method in this thesis: RBHS

All the machine learning based hot spot prediction methods mentioned above use a data matrix that contains protein sequence- and structure-based features. More information on the data matrix is given in Chapter 3. Such data matrices often contain values that can be corrupted by errors or noise that are caused due to experimental mistakes, computational tolerances and/or human errors [Can+11]. Such corruptions adversely affect the predictive ability of the current machine learning based hot spot prediction algorithms [GG19]. Predictive ability of an algorithm refers to its ability to predict test samples for the test data on which it has not been trained. In other words, predictive power is predicting correctly labels for test data that the model has not seen before. Therefore, using an approach where the data matrix contains reduced noise would be highly desirable.

In my method, namely, RBHS (**R**obust Principal Component Analysis-(RPCA) **B**ased prediction of protein-protein interaction **H**ot **S**pots) [Sit+21], this issue is addressed by pre-processing the data matrix using Robust Principal Component Analysis (RPCA) [Can+11]. RPCA is a variant of the traditional Principal Component Analysis (PCA) [WEG87; Jol02] method, and it is particularly useful for data matrices that may contain corrupted entries, such as the ones considered here. In reference [Can+11], the authors have showed that a noisy matrix D can be decomposed into a low rank matrix A that contains reduced noise and a sparse matrix S, regardless of the number of corrupted or missing entries (i.e. robustly). The low rank matrix obtained after applying RPCA is then the new data matrix for the developed pipeline in this thesis, RBHS, for identification of hot spots. After RPCA, feature selection is performed on the new less noisy low rank matrix A using gradient boosting methods and the selected features are passed on to a classifier that maps these features to hot spots and null spots. I apply this method, to curated benchmark datasets HB-34 and

BID-18 [LLD18]. Detailed information about these datasets will be provided in Section 4.2. Various classifiers, like Support Vector Machines (SVM), Random Forests (RF), Gradient Boosting Machines (GBM) and Extreme Gradient Boosting (XGB) are used in this thesis and their performance is compared through a number of performance metrics, such as, accuracy, sensitivity, precision, F1-score, and Matthews Correlation Constant (MCC). The performance of the RBHS pipeline is also studied through ROC (Receiver Operating Characteristics) curves and Precision-Recall curves. Hyperparameter tuning for the RPCA algorithm and the different classifiers is performed to select the best performing algorithm parameters and model for each of the classifier, respectively. Performance of the RPCA with different noise intensities is also studied. RBHS pipeline is compared with the unprocessed (raw) HB-34 and BID-18 data and with traditional PCA. After thorough experimentation, XGB and F1-score were found to be the best performing classifier and performance metric, respectively. The best performing model is named RBHS+XGB. The natural choice for F1-score also stems from the fact that the datasets are imbalanced, the number of null spot residues is much larger than the number of hot spot residues. F1-score is a metric that takes this imbalance into account by calculating the harmonic mean between precision and recall. RBHS+XGB was further compared with several state-of-the-art approaches for PPI hot spot prediction and was found to outperform many of them, as shown in Section 5.3.

1.5 Thesis Organization

The content of this thesis is organized as follows:

1. Chapter 2 describes in detail the biological relevance of the problem of prediction of hot spots. It describes in detail the biology of hot spots residues, protein-protein interactions, protein complexes and interfaces.

2. Chapter 3 gives a brief background of machine learning and the various terms and terminologies associated with machine learning used for hot spot prediction.

3. In chapter 4, I give a detailed description of my pre-processing pipeline RBHS and of all the machine learning methods I used for predicting hot spots. A workflow diagram for my method is added in this chapter for ease of understanding. Next, it contains the computational details of all the algorithms used. It also includes computational details of the experiments used for comparing RBHS with other techniques.

4. Chapter 5 includes results of comparing RBHS with other pre-processing algorithms, as well as comparing RBHS + Extreme Gradient Boosting classifier with other state-of-the-art methods for predicting hot spots. Finally, this chapter includes a detailed interpretation and analysis of all the results presented.

5. Chapter 6 draws conclusions from the method and results presented in this thesis, and also outlines future perspectives of my work.

1.6 List of publications

The method and results presented in Chapter 4 and Chapter 5 respectively of this thesis have been published in the following paper:
Divya Sitani, Alejandro Giorgetti, Mercedes Alfonso-Prieto, and Paolo Carloni. "Robust principal component analysis-based prediction of protein-protein interaction hot spots." Proteins: Structure, Function, and Bioinformatics 89, no. 6 (2021): 639-647, doi: 10.1002/prot.26047.

The data and the codes developed for this thesis have been added to the following GitHub repository:
https://github.com/Divya1205/RBHS_Sitani.

The conception of this work, data preparation, experiments, generation of results, analysis of the results, and writing of the manuscript was done by me, Divya Sitani. Prof. Dr. Alejandro Giorgetti, Dr. Mercedes Alfonso-Prieto and Prof. Dr. Paolo Carloni helped to review the manuscript.

2 Protein-protein interactions and Hot Spots

2.1 Proteins

Proteins are among the most abundant organic molecules in living systems and are very diverse in structure and function in comparison with the other classes of macromolecules. A single cell may contain thousands of different proteins, each with a distinctive function. Proteins have diverse functions because all proteins are made up of different arrangements of the same 21 amino acids. Even though, proteins have different structures and different functions, they are all polymers of amino acids arranged in a linear sequence and connected by peptide bonds (polypeptide chain).

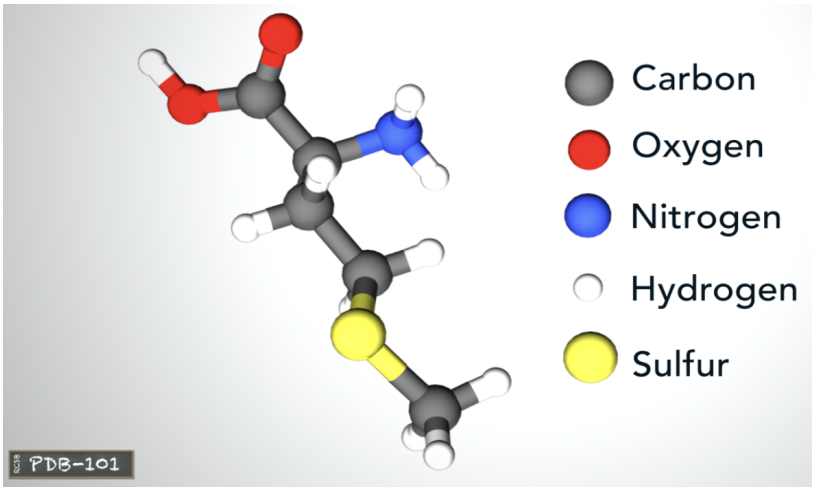


Figure 2.1.1: Chemical composition of an amino acid [PDB].

Amino acids are the monomers that make up proteins. Each amino acid has the same fundamental chemical structure that consists of a central carbon atom (C_α) attached to an amino group (NH_2), a carboxyl group (COOH) and a hydrogen atom. There is another atom or group of atoms attached to the central carbon atom, the R side chain and this side chain is the difference between the 21 amino acids.

The chemical nature of the amino acid within its protein is decided by the chemical nature of the R group. Hydrophobic or apolar amino acids have carbon rich side chains that do not interact well with water. Hydrophobic amino acids have the tendency of adhering to one another in aqueous environment. They are Alanine (Ala), Leucine

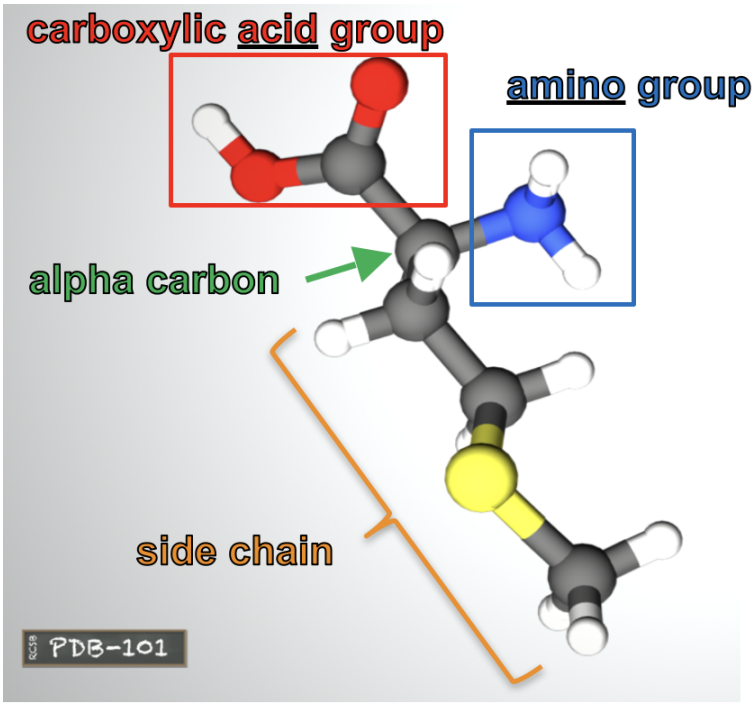


Figure 2.1.2: Structure of an amino acid [PDB].

(Leu), Phenylalanine (Phe), Valine (Val), Isoleucine (Ile), Methionine (Met), Proline (Pro) and Tryptophan (Trp) [Ber+00a; PDB]. Hydrophilic or polar amino acids interact well with water and are Serine (Ser), Threonine (Thr), Aspargine (Asn), Glutamine (Glu), Cysteine (Cys), Tyrosine (Tyr) [Ber+00a; PDB]. The charged amino acids interact with oppositely charged amino acids or hydrophilic amino acids and are Arginine (Arg), Histidine (His), Lysine (Lys), Aspartic Acid (Asp) and Glutamic Acid (Glu). The amino acid with no side chain is Glycine (Gly). Gly is hydrophobic in nature [Ber+00a; PDB]. This is summarized in Fig. 2.1.3.

Each amino acid is attached to another amino acid by a covalent bond, known as the peptide bond, that is formed by a condensation reaction. During protein synthesis, the carboxyl group of the amino acid at the end of the growing polypeptide chain reacts with amino group of an incoming amino acid to form a peptide bond by the elimination of water. Thus, a protein chain is formed by numerous amino acids in which the amino group of the first amino acid and the carboxyl group of the last amino acid stay intact and the chain extends from the amino to the carboxyl terminus. This chain is called a polypeptide chain or backbone, shown in Fig 2.1.4. Amino acids in a polypeptide chain are devoid of a hydrogen atom at the amino terminal and an OH group at the carboxyl terminal (with the exception of the ends). This is the reason why amino acids are also called amino acid residues, i.e. what remains after loss of a water molecule and formation of the peptide bond [Rye+16].

A protein is a polypeptide or polypeptides that have combined together and have a distinct shape and a unique function. After protein synthesis (translation), most

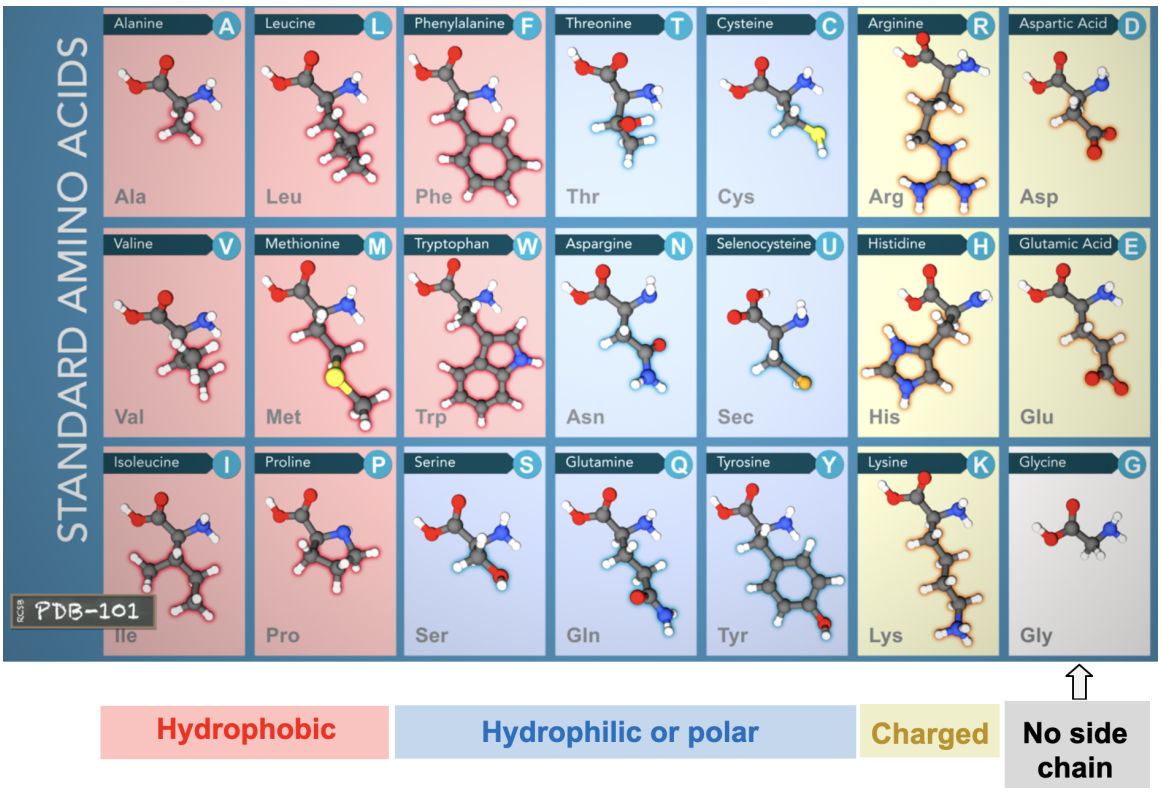


Figure 2.1.3: Classification of amino acids based upon the physicochemical properties of the side chains of amino acids [PDB].

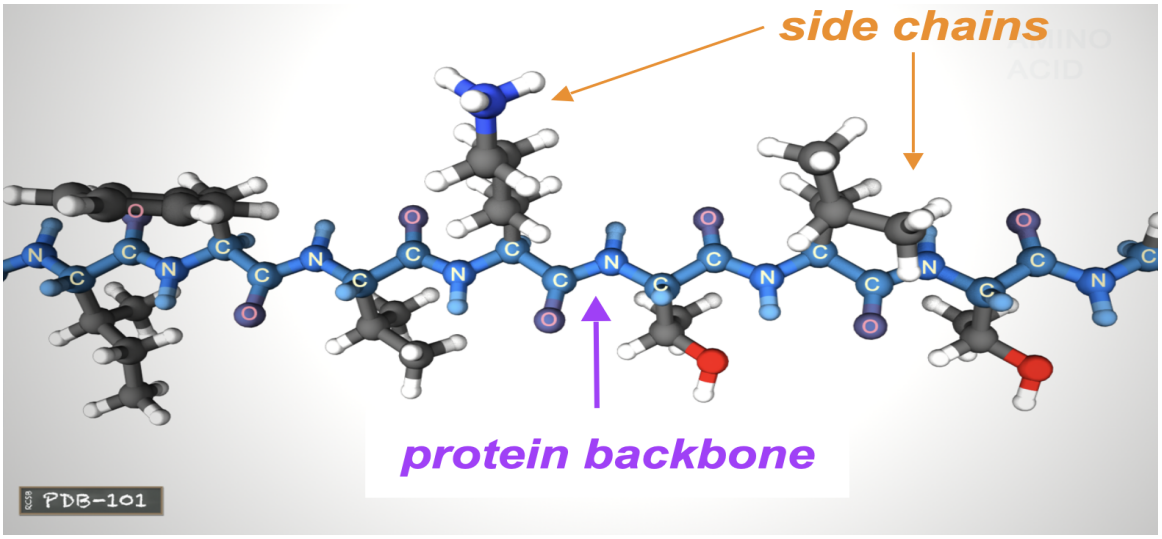


Figure 2.1.4: The linked series of carbon, nitrogen, and oxygen atoms make up the protein backbone and the protein side chains are hanging from it [PDB].

proteins are modified. These are known as post-translational modifications. They may undergo cleavage, phosphorylation, or may require the addition of other chemical

groups. The protein is fully functional after these modifications [Rye+16].

2.1.1 Levels of protein structure

The shape of a protein is critical to its function. To understand how a protein gets its final shape, it is important to understand the four levels of protein structure: primary, secondary, tertiary and quaternary.

2.1.1.1 Primary Structure

The primary structure of a protein is the unique linear sequence of amino acids in a polypeptide chain, and this sequence is determined by the gene that encodes the protein. The sequence defines how the protein will fold and thereby also determines the function of the protein. Any change in this gene sequence will result in a different polypeptide chain, and in turn result in a change in protein structure and function. For example, in the sickle cell anemia disease a single substitution in the amino acid sequence of hemoglobin causes the normally biconcave or disc-shaped red blood cells to assume a "sickle" shape, which clogs the arteries [PDB; MG15].

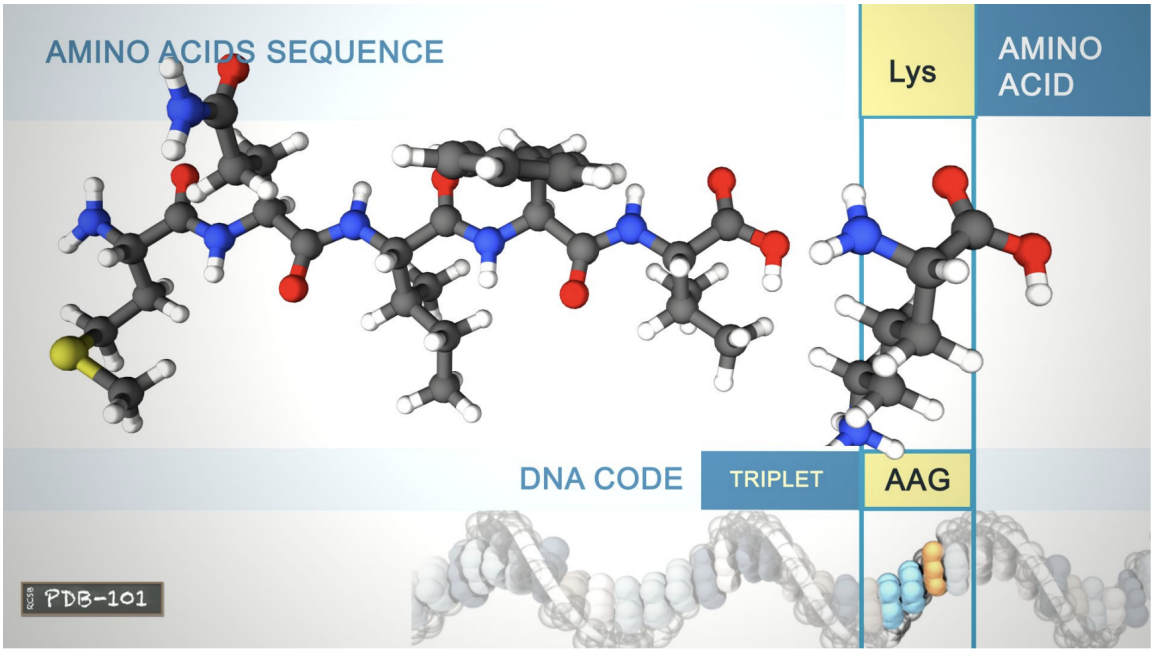


Figure 2.1.5: The primary structure of a protein is the linear sequence of amino acids, as encoded by the DNA genetic code [PDB].

2.1.1.2 Secondary Structure

The next level of protein structure is the secondary structure, that are locally folded structures that form within a polypeptide chain due to interactions between atoms of

the backbone excluding the side chains or the R groups.The protein chains often fold into two types of secondary structures: alpha (α) helices or beta (β) pleated sheets. These structures are held in shape by hydrogen bonds between carbonyl oxygen atom of one amino acid and amino hydrogen atom of another amino acid of the backbone. In the α helix (Fig. 2.1.5), the hydrogen bond forms between every fourth amino acid

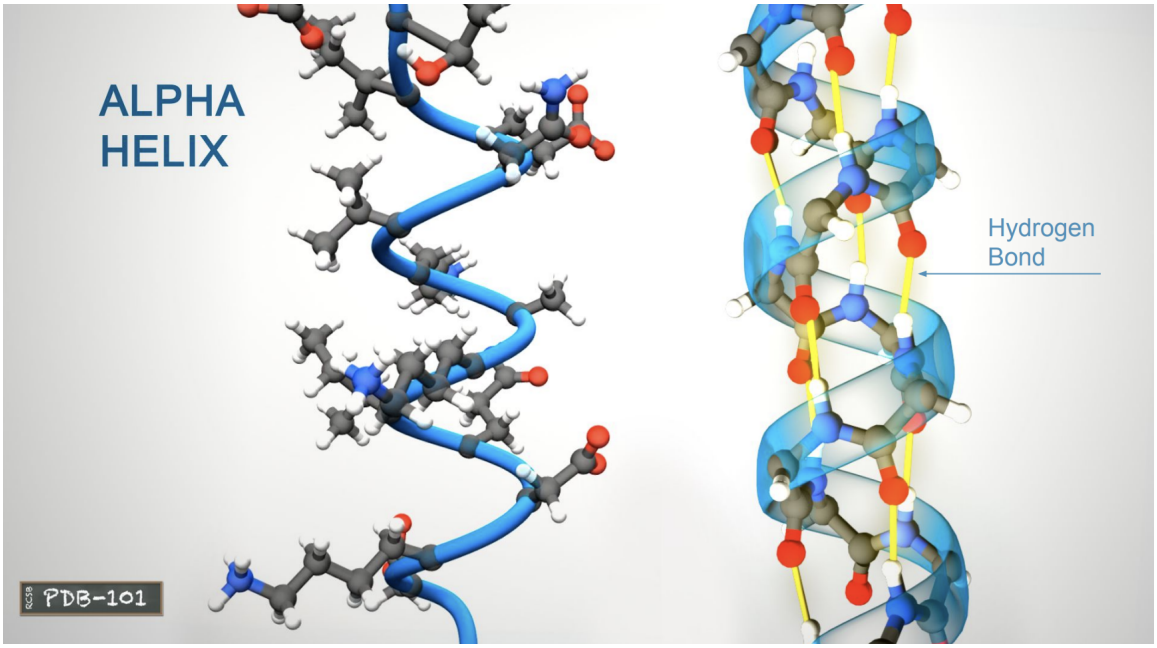


Figure 2.1.6: Secondary Structure: Alpha (α) helix [PDB].

and causes a twist in the amino acid chain, resulting into a helical structure. The R groups of the amino acid stick outward from this α helix. In the β pleated sheets (Fig. 2.1.6), multiple segments of a polypeptide chain line up next to each other and result in a sheet like structure held together by hydrogen bonds. These bonds are formed between carbonyl and amino groups of the backbone on adjacent β strands. The R groups extend above and below the plane of the sheet. The strands of a β sheet are either parallel (running in the same N- to C- terminal direction) or antiparallel (running in opposite N- to C- terminal directions) [PDB; BTS02].

2.1.1.3 Tertiary Structure

The tertiary structure of a protein is the complete three-dimensional shape of the polypeptide chain. This shape is caused by chemical interactions between various amino acids and regions of the polypeptide. Primarily, the interactions among R groups create the complex three-dimensional tertiary structure of a protein [PDB; BTS02]. For example, soluble proteins mainly form globular shapes with hydrophobic side chains sheltered inside, away from the surrounding water (Fig. 2.1.8), while membrane-bound proteins form hydrophobic residues that are clustered together on the outside, so that they can interact with the lipids in the membrane. In highly

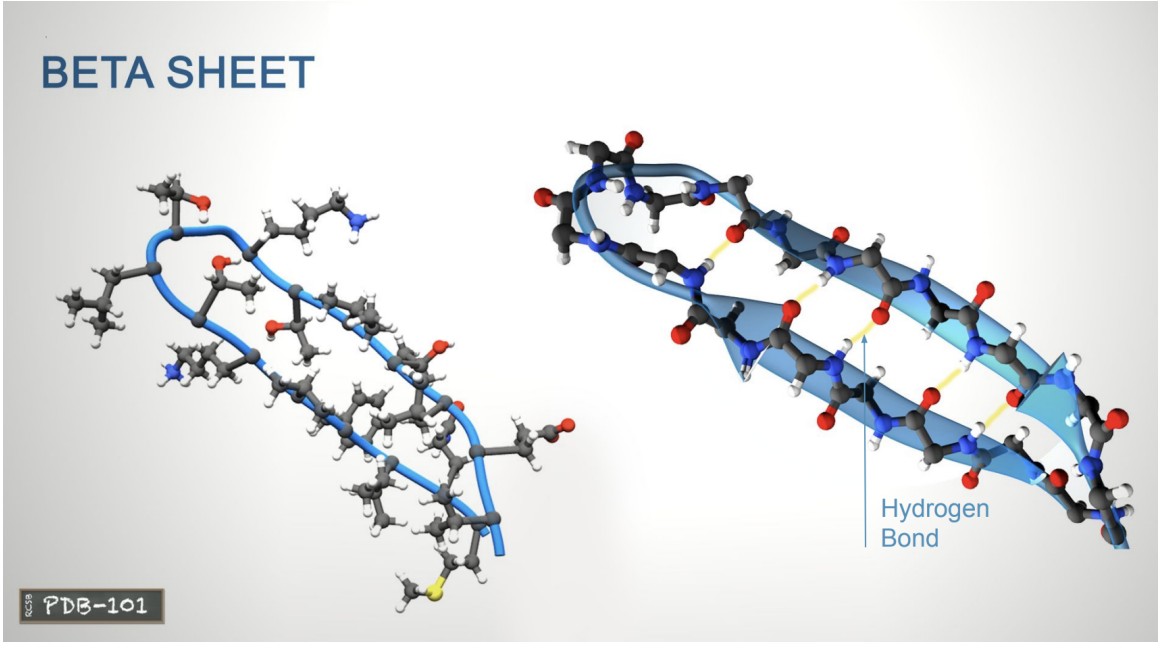


Figure 2.1.7: Secondary Structure: Beta (β) sheets [PDB].

charged proteins, charged amino acids on the surface allow proteins to interact with molecules that have complementary charges [PDB]. The functions of many proteins rely on their three-dimensional shapes. For example, hemoglobin forms a pocket to hold heme, a small molecule with an iron atom in the center that binds oxygen [PDB].

2.1.1.4 Quaternary Structure

Two or more polypeptide chains (subunits) can assemble together to form one functional molecule with several subunits. Quaternary structure is the number and arrangement of these protein subunits with respect to one another. If the final protein is made of two subunits, the protein is said to be a dimer. If there are three subunits together, the protein is called a trimer; four subunits make up a tetramer, and so on and so forth. If the subunits are identical, the prefix "homo" is used, as in "homodimer." If the subunits are different, we use "hetero," as in "heterodimer." For example, hemoglobin is a combination of four polypeptide subunits, two α and two β subunits. One α and one β subunit come together to form a heterodimer, and two of these heterodimers interact together to form one hemoglobin molecule (Fig. 2.1.9).

2.2 Protein-protein interactions

Proteins rarely act alone [Les10]. Instead, they interact with other proteins and biomolecules, which results in an intricate network of interactions. The complete set of protein-protein interactions in a living organism is known as the interactome. Protein - protein interactions are very essential for performing various biological processes

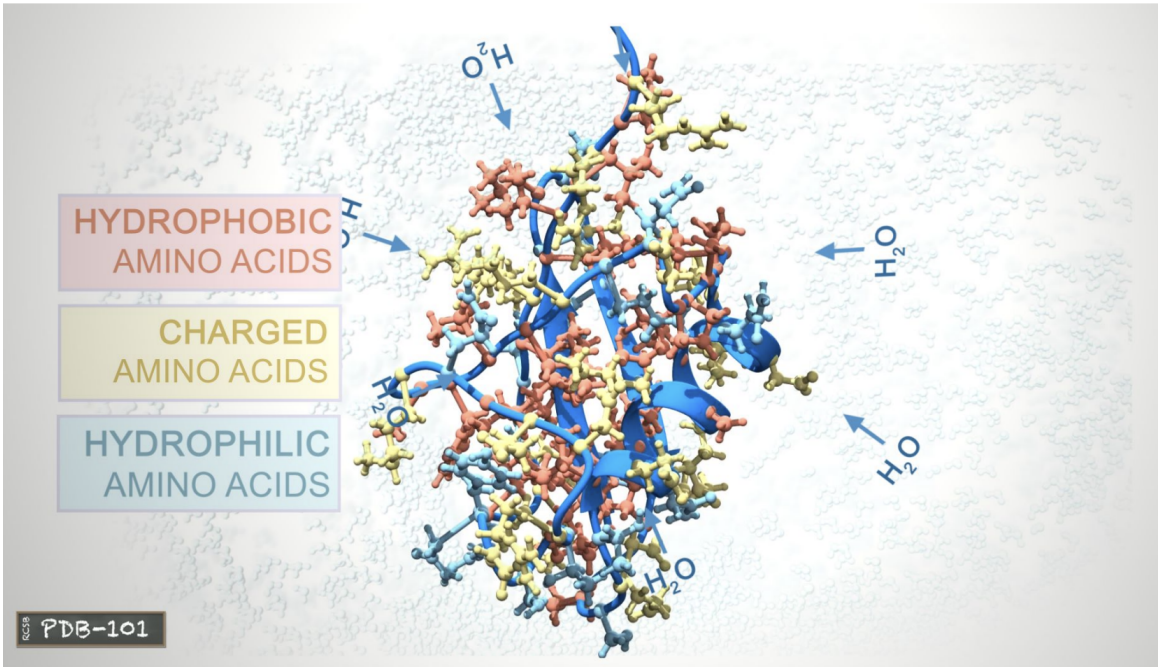


Figure 2.1.8: Tertiary Structure of a protein [PDB].

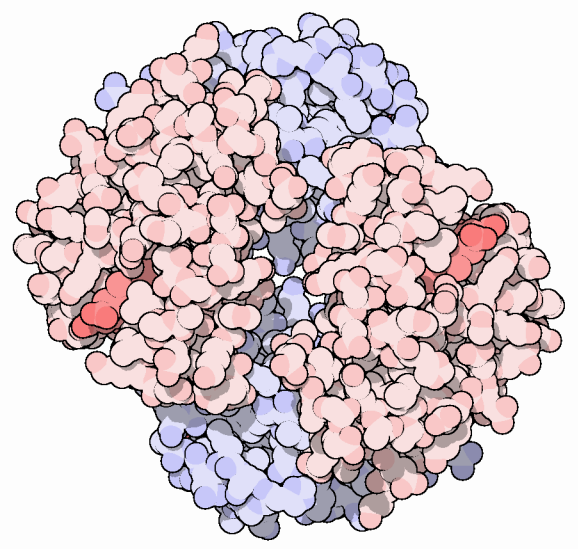


Figure 2.1.9: Hemoglobin molecule with its four polypeptide subunits. Heme is shown in red [PDB].

including cell to cell interactions, metabolism and developmental control, among others [Rao+14].

The non-covalent interactions between the R side chains are the basis of protein folding, protein assembly and also protein-protein interactions. These contacts take place under various conditions, and they make multiple interactions and associations within and between proteins. For example, protein structure is determined through interactions between residue side-chains. In this instance, the interactions are permanent because they typically last for the lifetime of a protein. However, non-covalent residue–residue interactions can also be transient, like in receptor–ligand interaction or in signal transduction. These interactions last for only short times. Thus, the transient interactions form signaling pathways, whereas the permanent interactions form a stable protein complex [Rao+14; OR03].

2.3 Protein-protein complexes

Nearly 80% of proteins operate in complexes [Rao+14]. A protein complex is a group of polypeptide chains linked by non-covalent protein-protein interactions (PPIs). As mentioned earlier, nearly all biological processes involve protein-protein interactions and quite a lot of the processes require multiple protein-protein interactions to form the quaternary structure of multimeric proteins, thus, forming the protein-protein complexes. One particular protein can be involved in a variety of protein complexes. The same complex can perform different functions depending on multiple factors like the stage of cell cycle, the nutritional status of the cell, the cellular compartment, etc. The understanding of protein-protein interactions at atomic detail requires the knowledge of the three-dimensional structure of protein complexes and protein-protein interfaces.

Proteins interact with each other through their interfaces. Protein–protein interfaces are non-uniform surfaces where two proteins make direct physical contact [GS10]. PPI occurs through interactions between residues on two opposite interfaces [DS15]. Interfaces consist of interacting residues that belong to two different chains, along with residues in their spatial vicinity. Thus, interfaces consist of fragments of each of the individual chains and some isolated residues (Fig. 2.3.1).

Mutations can disrupt a protein interface by modifying its physicochemical, structural, and energetic characteristics [DS15]. Moreover, disease-causing mutations are expected to have a greater impact on protein structure, function, and protein complex thermodynamics when occurring in interface residues [Ten+09]. Even though a protein interface may occupy a large area, only a small subset of its buried residues plays a crucial role in the binding free energy of the complex. These key energetic residues are known as hot spots. The experimental method to identify hot spots is Alanine Scanning Mutagenesis [Wel91].

Alanine (Ala) scanning is a widely used mutagenesis approach in which residues in a target protein complex are systematically substituted for Ala at selected positions by site-directed mutagenesis. Ala is used as it is a non-bulky, chemically inert

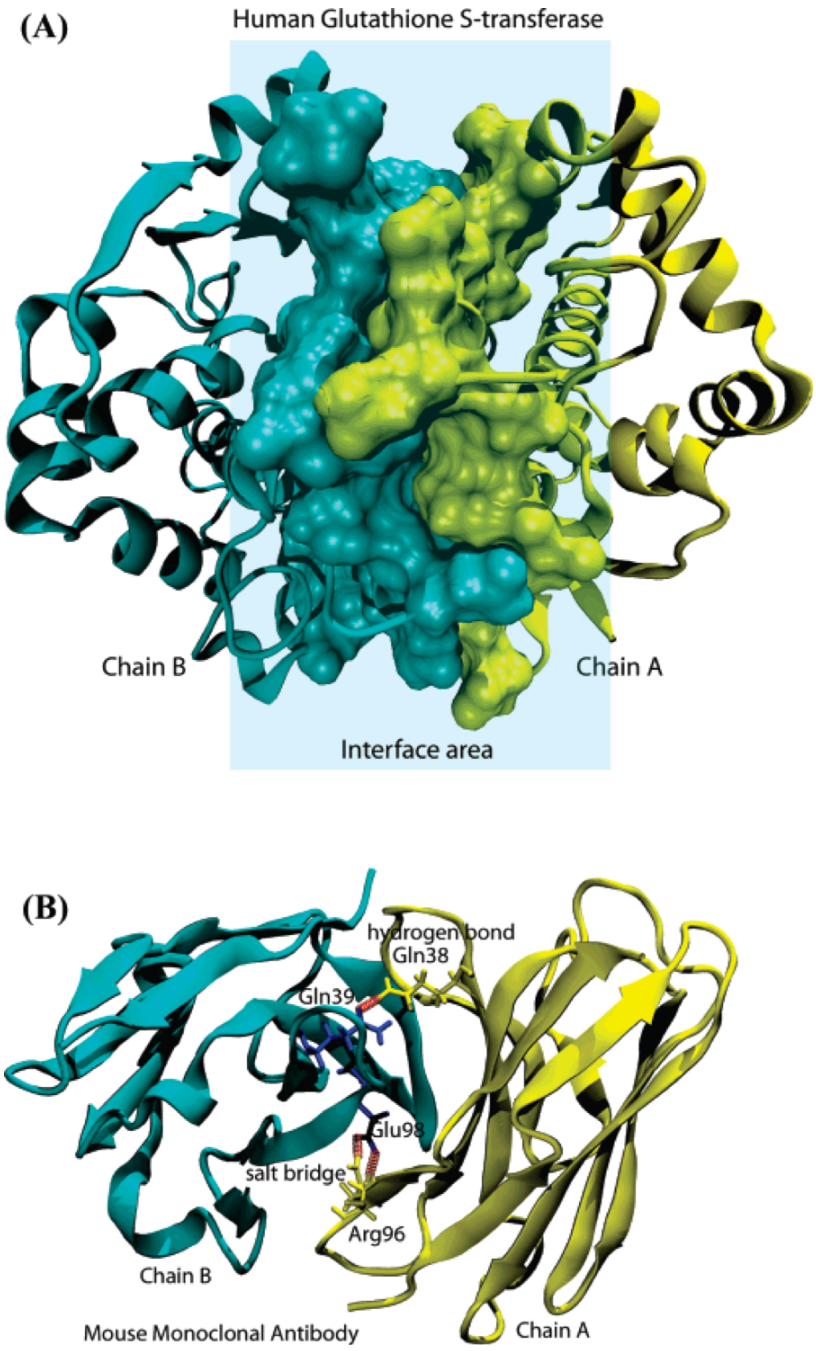


Figure 2.3.1: (A) represents the complex human Glutathione S-Transferase, PDB ID: 10GS, Chains A and B. The interface has been shown with surface representation whereas the rest of the protein in ribbon representation. (B) represents a few interface residues along with details of the respective non-covalent residue interactions of the interface of Mouse Monoclonal Antibody D1.3 (PDB ID: 1KIR, Chains A and B) [Kes+08].

residue and its methyl functional group nevertheless mimics the secondary structure preferences that many other amino acids possess. Substitution with Alanine residues eliminates side-chain interactions without altering main-chain conformation or introducing any steric or electrostatic effects. Thus, it is often the preferred choice for testing the contribution of specific side-chains while preserving the native protein structure [MFR07; Wel91].

2.4 Protein-protein interaction hot spots

The application of Alanine Scanning Mutagenesis (ASM) to protein-protein interfaces helped to discover a highly uneven distribution of energetic contributions of individual residues across each interface, and that only a few key residues do contribute significantly to the binding free energy of protein–protein complexes: the so-called "hot spots". Hot spots have been defined as those sites where Alanine mutations cause a significant change in the binding free energy of at least 2.0 kcal/mol. In a nutshell, experimental Alanine Scanning Mutagenesis identifies hot spots experimentally by systematically mutating each interface residue to Alanine and measuring the change in $\Delta G_{binding}$ ($\Delta\Delta G_{binding}$=ΔG_{mutant}− $\Delta G_{wild\text{-}type}$). If ($\Delta\Delta G_{binding}$) ≥ 2.0 kcal/mol, the interface residue is defined as a hot spot, otherwise as a “null spot” [MFR07; CW89; BT98]. Energetic hot spots tend to be enriched in disease-causing mutations compared to non-hot spots [DS15].

The discovery of hot spots in PPI interfaces, made it possible to target a broader range of PPIs with small molecule drugs. Drug molecules interacting with these hot spots may interfere with PPIs and the downstream pathways they influence [Pet+16b; Sco+16].

Experimental ASM is expensive, time-consuming, and labor-intensive, so the available data on hot spots is limited [BT98]. Thus, there has been a major increase in the use of *in silico* methods to identify hot spots. Some of these methods rely on energy based scoring functions (like FOLDEF [GNS02] and Robetta [KB02]) or molecular dynamic simulations [MK99; HMK02; GF08; Bre+09]. However, these approaches are computationally expensive and therefore difficult to apply in a high-throughput mode.

Machine Learning (ML) (subset of Artificial Intelligence (AI)) based methods have been extensively used to identify hot spots in the past years. Machine learning has been used in biology for a number of decades, but it has steadily grown in importance to the point where it is used in nearly every field of biology. However, only in the past few years has the field taken a more critical look at the available ML strategies and begun to assess, which methods are most appropriate in different scenarios, or even whether they are appropriate at all. The two major goals of using machine learning in biology is to make accurate predictions for tasks where enough experimental data is not available, and the second is to interpret the results of ML algorithms with respect to the biological context of the problem at hand. A short review of machine learning methods that are used for the prediction of hot spots is given in the next chapter.

These methods were also used for the prediction of hot spots in this thesis.

3 Basics of Machine Learning

Machine Learning is the field of study that gives a computer the ability to learn without being explicitly programmed [Sam59; Mit17]. The aim is to develop methods for computers to "learn" from available data. The learning process [Mit17] can then be defined as a method through which a computer is able to learn to perform tasks, T such that given experience, E relevant to T, improves its performance as measured by a performance metric P. The experience E is mostly available in the form of a dataset that contains typical input data, the task T could be one of classification, regression, synthetic data generation or many others where the relation between E and T is complex and cannot be analytically defined. Broadly, a machine learning method involves learning of an adaptive model, which maps the available input data to a set of output variables, for a given task [Mit17; Bis06]. The model is adaptive in nature because the it learns through the experience. The learning algorithm is provided with input data, which is also known as the *training data*, and it learns an adaptive model based on the given task. This is known as the *training* phase. Once the model is trained or learnt it can be used to do *inference*, i.e., perform the given task on new data previously unseen by the model. This is known as the *testing* phase. The dataset used during inference is known as the test dataset. Depending on the nature of the input data and the given task, the learning process can be accomplished in a number of ways, namely, supervised learning, unsupervised learning, and reinforcement learning.

3.1 Types of Machine Learning

Supervised Learning

In supervised learning, the learning algorithm is provided with training data that includes the input data and the corresponding target output variable for the given task. The model learns a pattern or mapping between the input and the target variable. During testing, the target variables are withheld and inference is performed on the test input data to generate the target output variables from the model. The predicted target output variable is compared with the actual or *ground truth* output variable that had been withheld to test the predictive power of the model [Bis06].

In a supervised learning setting, the dataset, both training and test, consists of examples $\{(x_i, y_i)\}_{i=1}^{N}$. Each x_i is an input example and is called a *feature vector*. Each feature vector is of dimension, D and each dimension $j = 1, 2, ..., D$ contains numerical information that is an individual measurable property or characteristic of the observed phenomenon [Bis06]. There are N feature vectors because there are N

examples, and these are concatenated to form a data matrix of size $D \times N$. The target output variable, y_i, in most cases, is an element belonging to a finite set of integers or a real value. Based upon the type of value of the output variable, supervised learning is of two types:

1. **Classification** involves predicting a class label, i.e., a set of integers.
2. **Regression** involves predicting a numerical label, i.e., a real value.

Unsupervised Learning

Unsupervised learning allows learning of a model without any explicit target output variables. The dataset only consists of input examples without any class labels. The unlabeled dataset of input examples or feature vectors can be written as $\{x_i\}_{i=1}^{N}$. Thus, there is no instructor or teacher and the algorithm must learn to make sense of the data without any guide [GBC17]. The goal of an unsupervised learning algorithm is to create a model that takes these feature vectors as input and identifies patterns or groupings in the data or transforms x into another vector, which makes it easier to learn for a specific task. Depending upon the application, unsupervised learning is of three types:

1. **Clustering** involves finding groups in the unlabeled data based on their similarities or differences.
2. **Density Estimation** involves summarizing the distribution of the data. Thus, it assumes that sample data comes from a population that follows a probability distribution based on a fixed set of parameters.
3. **Dimensionality Reduction,** where the output of the model is a feature vector that has less number of features than the input feature vector x.

Straddling the boundary between supervised learning and unsupervised learning, is an area of machine learning called *semi-supervised learning* [CSZ10; ZG09]. Semi-supervised learning is concerned with learning a model where the number of labeled data is small, but a large corpus of unlabeled data is available. This technique is capable of generalizing to new unseen data, which is known as inductive learning, and also to the unlabeled training (available) data, which is known as transductive learning.

Reinforcement learning

Reinforcement learning is a type of machine learning where the machine“lives" in an environment and the state of the environment is perceived as a vector of features by the machine. The machine performs actions in every state and different actions bring different rewards and the machine must learn to maximize the reward. The machine

uses a feedback loop between learning and the experience. In many complex domains, reinforcement learning is the only feasible way to train a program to perform at high levels. For example, in game playing, it is very hard for a human to provide accurate and consistent evaluations of large numbers of positions, which would be needed to train a model directly from examples (experience). Instead, the program can be told when it has won or lost, and it can use this information to learn a model that provides reasonably accurate estimates of the probability of winning from any given position [RP15]. Impressive recent results include the use of reinforcement learning in Google's AlphaGo in out-performing the world's top Go player.

In this thesis, classification is the machine learning task that will be studied. In the following section, the process of learning a model to classify given data and the performance evaluation of the model, or classifier, is discussed in detail.

3.2 Classification

Supervised learning is the most common and frequently used type of machine learning. Among various tasks classification is one of the most studied and in this thesis as well the task of classification will be used extensively. The main steps for a classification task in a supervised learning workflow are the following:

1. Data acquisition
2. Data preparation and preprocessing
3. Data splitting
4. Classification algorithms
5. Performance metrics
6. Testing the learned classifier

The various steps in the above workflow are discussed in detail next.

3.2.1 Data Acquisition

The process of gathering data depends on the domain of the problem or task we want to solve. The data set can be collected from various sources such as a file, database and many other similar sources. In general, studies combine data from more than one database and filter the redundancies. Widely used databases of experimental verified hot spots include the Alanine Scanning Energetics Database (ASEdb) [TB01], the Binding Interface Database (BID) [Fis+03], the Protein-protein Interaction Thermodynamic (PINT) database [TB01] and the Structural database of Kinetics and Energetic of Mutant Protein Interactions (SKEMPI) [TB01]. These databases are

further explained in detail in Chapter 4. During data acquisition, each tool and technique is sensitive to numerical and human precision, and this introduces systemic and gross errors or noise into the collected dataset. Thus, the collected data cannot be used directly for performing classification as there might be missing data, extreme values, unorganized text data or noisy data. Therefore, one must perform a data preparation and preprocessing step before a classifier can be learnt.

3.2.2 Data Preparation and Preprocessing

In data preparation, feature vectors (input) and the target variable (output) are identified as described in Section 3.1. For this task, tools and techniques pertaining to the specific domain are used to encode features for the given samples in the database. This generally leads to representation of data in the form of matrix of size $D \times N$, where D is the size of the encoded feature and N are the number of data samples. This makes the dataset amenable to computational techniques. The dataset in its current form may contain features with different numerical scales, missing data or noisy data. A final dataset, that can be used by machine learning algorithms, is prepared by preprocessing the dataset, i.e., the data matrix. Depending upon the kind of errors present in the dataset, different preprocessing is required.

3.2.2.1 Missing Values

In some cases, some values are missing for some features in the dataset [Has+09]. This happens due to human errors where the person forgets to fill some values or did not measure the value at all. The typical methods of dealing with missing value problems include:

1. Removing the samples that have features with missing values. This is not a very feasible idea if the dataset is not large enough.

2. Using imputation methods where the missing data is replaced with substituted values. For example:

   a) **Imputation** [LEA15]: It is the process of replacing the missing values with a representative value from the dataset. Imputation can be single or multiple in nature. Single imputation involves replacing the missing value once with a representative value, such as the mean or median of the feature value. Other techniques for single imputations are random imputation, last observation carried forward (for time series) and the like. Multiple imputation involves replacing the missing value multiple times by generating plausible values by modeling the distribution of the features in the observed data.

   b) **kNN (k-Nearest Neighbors)**: Fill data with a value from other examples or neighbors that are similar in respect to a certain distance metric.

3.2.2.2 Feature Encoding

Some learning algorithms work with only numerical feature vectors. For example, when some feature in the dataset is categorical, like colours or pets, we can transform these variables into numerical values [Bur19]. Popular feature encoding techniques are:

1. **One Hot Encoding**: Convert all unique values into lists of 0's and 1's where the target value is 1 and the rest are 0's. For example, when the color of a green, red, and blue needs to represented, the feature for a green car would be $[1, 0, 0]$ and a red one would be $[0, 1, 0]$.
2. **Label Encoder**: Convert labels into distinct numerical values. For example, if your target variables are different animals, such as dog, cat, bird, these could become 0, 1, and 2, respectively.

3.2.2.3 Feature Scaling

Feature scaling is the method of converting the different numerical scales of the various features into a standard range. These methods are also known as feature normalization. Feature scaling is commonly done when the values of different features in the data matrix have different scales [Bur19]. The two most popular feature scaling methods are *min-max* normalization and *mean-variance* normalization or standardization.

Min-max normalization is the method of converting the range of values that a numerical feature can take, into a standard range of values, typically in the interval $[-1, 1]$ or $[0, 1]$. The min-max scaled feature, $\bar{x}^{(j)}$ can be written as:

$$\bar{x}^{(j)} = \frac{x^{(j)} - min^{(j)}}{max^{(j)} - min^{(j)}} \tag{3.2.1}$$

where $min^{(j)}$ and $max^{(j)}$ are the minimum and maximum value of the feature j in the dataset, respectively.

In Standardization (or z-score normalization) [Bur19] the feature values are rescaled so that they have the properties of a standard normal distribution with $\mu = 0$ and $\sigma = 1$ (Fig. 3.2.1). Here, μ is the mean or the average value of the feature, averaged over all examples in the dataset and σ is the standard deviation from the mean.

$$\bar{x}^{(j)} = \frac{x^{(j)} - \mu}{\sigma} \tag{3.2.2}$$

3.2.2.4 Dimensionality Reduction

In many real-world machine learning problems, the number of features, D is large compared to the number of available data samples N. Sometimes, many of these features are correlated or redundant. Moreover, the higher the number of features, the harder it gets to visualize the dataset and then analyze or process it. As the dimensionality increases, the computational cost also increases, usually exponentially.

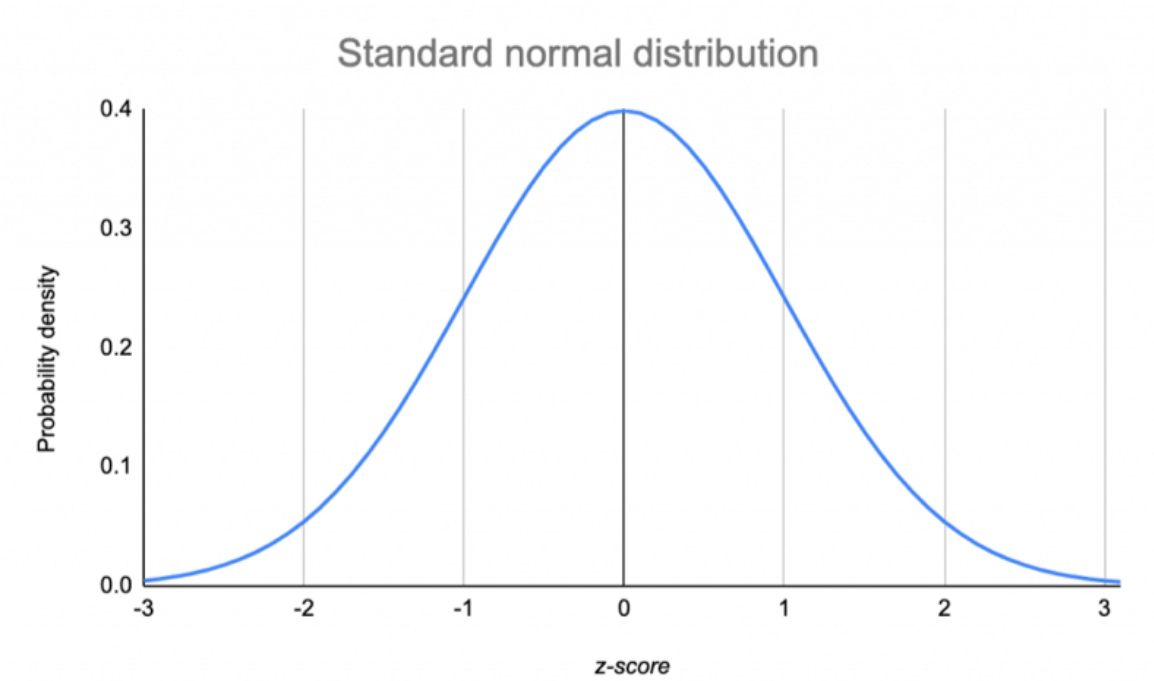
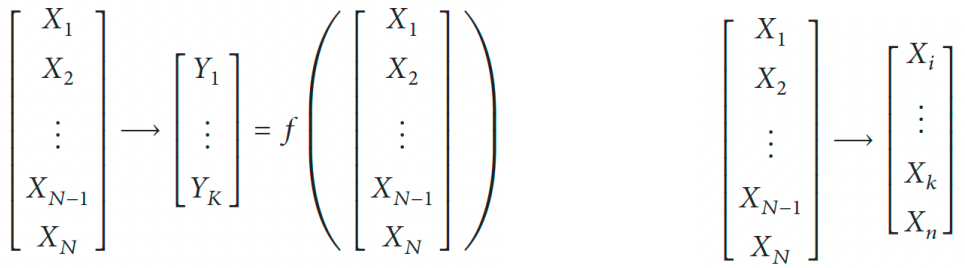


Figure 3.2.1: Standard Normal Distribution with $\mu = 0$ and $\sigma = 1$.

This increase in difficulty of processing data with increase in the dimensions is known as the *curse of dimensionality* [Bis06]. Dimensionality reduction is the process of reducing the number of features under consideration, by obtaining a set of principal or important features. It is used to transform data with a large number of features (or dimensions) into a lower dimensional while preserving the different relationships between the data points. For example, data points that are similar (for example, two homologous protein sequences) should also be similar in their lower- dimensional form, whereas dissimilar data points (for example, unrelated protein sequences) should remain dissimilar [NH19; Moo+19; Gre+22]. Dimensionality reduction techniques can include both linear and non-linear transformations of the dataset [Gre+22]. Dimensionality reduction techniques can be categorized in two major ways:

1. **Feature Extraction**: It reduces the number of features in a dataset by creating new features from the existing ones and discarding the original features.This is shown in Fig. 3.2.2a. The new set of features will have different values as compared to the original feature values. The main aim is that fewer features will be required to capture the same information. Commonly used feature extraction techniques are principal component analysis (PCA) [WEG87; Jol02], Uniform Manifold Approximation and Projection (UMAP) [MHM20] and t-distributed Stochastic Neighbour Embedding (t-SNE) [VH08]. The technique to employ is dependent on the problem being solved: PCA retains original relationships between data points and is interpretable because each component is a linear combination of features of the original dataset. The applications of PCA in



(a) Feature Extraction (b) Feature Selection

Figure 3.2.2: Dimensionality Reduction Techniques

biological applications such as molecular dynamics trajectory analysis can be found in [Ste+06]. t-SNE strongly preserves local relationships between data points and is a flexible method that can reveal structure in complex datasets. Applications for t-SNE include single-cell transcriptomics [KB19; Gre+22].

There are some extensions of PCA that were developed to overcome the limitations of PCA. For example, by its very nature, PCA is sensitive to the presence of outliers and therefore also to the presence of gross errors in the datasets. This has led to attempts to define robust variants of PCA, and Robust principal component analysis (RPCA) has been used for different approaches to alleviate the limitations of PCA [Can+11]. Kernel PCA is an extension of PCA that allows for the separability of nonlinear data by making use of kernels. The basic idea behind it is to project the linearly inseparable data onto a higher dimensional space, where it becomes linearly separable [Mik+98].

2. **Feature Selection** Feature selection is the process of selecting a subset of relevant features from all the available features for use in model learning. In contrast to feature extraction techniques, feature selection techniques do not alter the original representation of the features, but merely select a subset of them (Fig. 3.2.2b). Thus, they preserve the original semantics of the features, hence, offering the advantage of interpretability by a domain expert [SIL07; GE03]. In general, there are three types of feature selection methods:

   a) **Filter Methods** apply a statistical measure to assign a score to each feature. The features are ranked by the score and then, depending upon the score value, it is decided if a particular feature will be selected or removed from the dataset. Afterward, this subset of features is presented as input to the classification algorithm (Section 3.2.4). These methods are often univariate. This means that each feature is considered separately, thereby ignoring feature dependencies, which may cause worse classification performance when compared to other feature selection techniques [SIL07]. Some examples of filter methods include the Chi squared test, information gain and correlation coefficient scores.

   b) In **wrapper methods**, a subset of features is used to train a model. Based

on the inferences drawn from the model performance at the previous step, it is decided whether to add or remove features from the feature subset. Thus, the feature selection problem is reduced to a search problem [SIL07]. Some common examples of wrapper methods are forward feature selection, backward feature elimination, and recursive feature elimination.

i. *Forward selection* is an iterative method that starts with having no feature in the model. In each iteration, the feature which improves the model performance keeps getting added on each iteration, till an addition of a new feature does not improve the performance of the model [HG15].

ii. *Backward elimination* starts with all the features and removes the least significant feature based on the p-value at each iteration. This is repeated this until there is no further improvement on the removal of features [KJ97].

iii. *Recursive feature elimination* is an example of backward feature elimination, which works by searching for a subset of features by starting with all features in the dataset and successfully removing features until the desired number remains. This technique starts by using all the features to train the model, ranking features by importance, discarding the least important features, and re-fitting the model. This process is repeated until a specified number of features remains. Features are scored either using the provided machine learning model (e.g. some algorithms like decision trees offer importance scores) or by using a statistical method [Guy+02].

c) **Embedded methods** learn which features best contribute to the accuracy of the model while the model is being created. The most common type of embedded feature selection methods are regularization methods [SIL07]. Regularization methods also called penalization methods introduce additional constraints into the optimization function of a predictive algorithm like a regression algorithm. Examples of regularization algorithms are the LASSO (Least Absolute Shrinkage and Selection Operator) [Tib11], Elastic Net [ZH05] and Ridge Regression [McD09].

An interesting thing to note here is that, with sufficient domain knowledge, one can have a clear understanding of the task and make conclusions as to why the model selects or removes certain features. In other words, why certain features are more important than the others. Feature selection thus aids explainability of the predictions generated by the machine learning algorithms.

3.2.3 Data Splitting

The preprocessed dataset is usually split into:

1. **Training set**, which is the set of samples used for learning the parameters of the model. This is generally 80 % of the total dataset.

2. **Validation set**, usually 10 % of total dataset. This dataset is used during training to estimate the performance of the trained model. It acts as the test set during training. The total data available for training includes the training set and the validation set. Though the validation set is considered part of the training data, but it is not used to specifically train the model. The trained model is validated on the validation set and, if performance of the model is not acceptable, then training is resumed. Thus, model hyperparameters, which define the model itself, are tuned on this validation dataset. Mostly, k-fold cross validation is used to split the total available data for training into training and validation set. The total available training set is split into k evenly sized partitions (common values being 5 or 10) to form k different training and validation sets, and the performance is compared across each partition to select the best hyperparameters that perform best across all partitions. K-fold cross validation helps to prevent overfitting of the model on the training data. An overfitted model will produce excellent results on training data, but will produce poor results on unseen data [Gre+22].

3. **Test set** (usually 10 % of total data): Model's final performance is evaluated on this set. This dataset should not be used to tune the model. The test set helps us to establish the quality of the learnt model and provides an insight into the generalization capability of the model, i.e., how well the model performs on similar data that it did not see during training. A test set should be independent of the training and validation dataset to ensure unbiased evaluation of the model during testing. If a data sample is present in both the training and test set, then a model can learn an identity mapping and still have very good performance. The test set is also called the 'hold-out set'and is used to assess the performance of the model on data not used for training or validation and gives an analysis of the model's expected real-world performance. The test set should be used only once, at the very end of the study [Gre+22].

Thus, construction of the three datasets play a crucial role in the learning process. They represent the experience, E that enables the model to learn the task, T that improves the performance, P. Already having discussed how the experience for the model can be created, the next section discusses how models can be learnt for the task of classification.

3.2.4 Classification Algorithms

Machine learning algorithms are described as learning a target function (f) that best maps input variables (X) to an output variable, (Y) i.e. $Y = f(X)$. This is a general learning task called predictive modeling where one would like to make predictions in the future (Y) given new examples of input variables (X) and the aim is to make

as many accurate predictions as possible. Here, X is the set of data points that are represented by a vector of features and the output Y is often called labels or classes or categories or targets. Class labels are often string values and must be mapped to numeric values before being provided to an algorithm for modeling. This is often referred to as label encoding, where a unique integer is assigned to each class label, as mentioned in §3.2.2.2. For example, the labels can be 0 and 1, when there are two classes, such as in hot spot prediction (Chapter 4).

Classification is defined as a process of assigning new observations into previously defined classes. It is a predictive modeling problem where a class label is predicted for a given example of input data. Furthermore, classification is a supervised learning problem, which means that it requires a training dataset with examples of inputs and their respective output labels to learn. As such, the training dataset must be sufficiently representative of the classification problem and have sufficient examples of each class label [Has+09]. Many different types of classification algorithms are available for modeling a classification predictive modeling problem and as such, there is no good theory on how to select which classification algorithm for a particular problem. The practical way is generally to use controlled experiments to find out which algorithms and their respective configurations result in the best performance for a given classification task. The classifiers used in this thesis are discussed next.

3.2.4.1 Support Vector Machine (SVM)

SVM is a supervised machine learning algorithm that is widely used for classification problems [Bis06; Guy+02; BGV92]. The main idea of SVM is to find the optimal separation boundary, or boundaries, between classes with the help of training data. In SVMs, these boundaries are called hyperplanes, which are identified by locating support vectors (or the instances that most essentially define classes) and their margins, which are the lines parallel to the hyperplane defined by the shortest distance between a hyperplane and its support vectors. The main idea is that, with a high enough number of dimensions, a hyperplane separating a particular class from others can always be found, thereby delineating dataset member classes. When repeated a sufficient number of times, enough hyperplanes can be generated to separate all classes in n-dimensional space. Importantly, SVMs do not look just for any separating hyperplane, but for the maximum-margin hyperplane, i.e. the one that resides equidistant from the respective class support vectors.

In feature space, when data is linearly separable, many separating hyperplanes can be chosen to identify classes, with the following form,

$$f(x) = w^T \cdot x + b = 0. \tag{3.2.3}$$

Here, W is a vector of weights, b is a scalar bias and $X \in \mathbb{R}^n$. Finding the maximum-margin hyperplane, or the hyperplane that resides equidistant from the support vectors, is done by using a Lagrangian formulation and the Karush-Kuhn-Tucker condi-

tions. The maximum-margin hyperplane can be expressed in the following form:

$$f(x) = b + \sum_{i=1}^{n} \alpha_i y_i x(i) \cdot x. \tag{3.2.4}$$

where b and α_i are learned parameters, n is the number of support vectors, i is a support vector instance, y_i is the class value of a particular training instance of vector $x(i)$ and x is a test point for which prediction needs to be made. In Equation 3.2.4 the dot product between $x(i)$ and x is calculated. Thus, once the maximum-margin hyperplane is identified and training is complete, only the support vectors are relevant to the model, as they define the maximum-margin hyperplane; all the other training instances can be ignored.

When data is not linearly-separable, the data is first transformed into a higher dimensional space using kernel functions, and this space is then searched for the hyperplane that can separate the samples. This is another quadratic optimization problem, in which the hyperplane can now be expressed by:

$$f(x) = w^T \cdot \phi(x) + b. \tag{3.2.5}$$

Here, $\phi(x)$ is a kernel function.

3.2.4.2 Ensemble methods

Machine learning models can be fitted to data individually or combined in an ensemble. An ensemble [Has+09] is a combination of simple individual models that together create a more powerful new model. *Boosting* is a method for creating an ensemble. It starts by fitting an initial model (e.g. a tree or linear regression) to the data. Then a second model is built that focuses on accurately predicting the cases where the first model performs poorly. The combination of these two models is expected to be better than either model alone. Then this process of boosting is repeated many times. Each successive model attempts to correct for the shortcomings of the combined boosted ensemble of all previous models. Examples of boosting algorithms are Adaptive Boosting (ADABoost), Extreme Gradient Boosting (XGB), Gradient Boosting Machines (GBM) [Has+09] among others.

Bagging [Has+09] is another ensemble method where the objective is to create several subsets of data from training data by choosing samples randomly with replacement. Each collection of subset data is used to train their own model. Thus, an ensemble of different models is obtained. The final output is based on majority voting after combining the results of all models. The combination of results from different classifiers is used, which is more robust than a single classifier. An example of bagging ensemble algorithm is Random Forest.

In this thesis three ensemble models are used, namely, random forests, gradient boosting machines, and extreme boosting machine.

1. **Random Forests (RFs)** [Bre01] is a supervised learning algorithm used for both classification and regression problems. Random forest is an ensemble learning method where a group of decision trees are combined to build a better model. A decision tree [Has+09] builds iteratively by asking questions to partition data at each step. A decision tree has three components: decision nodes, leaf or terminal nodes and a root node, Fig. 3.2.3. The decision tree divides a training dataset into branches, which further segregate into other branches. This sequence continues until a leaf node is attained. The leaf node cannot be segregated further. The nodes in the decision tree represent features that are used for predicting the outcome. Decision trees classify the examples by sorting them down the tree from the root to some leaf/terminal node, with the leaf/terminal node providing the classification of the example.

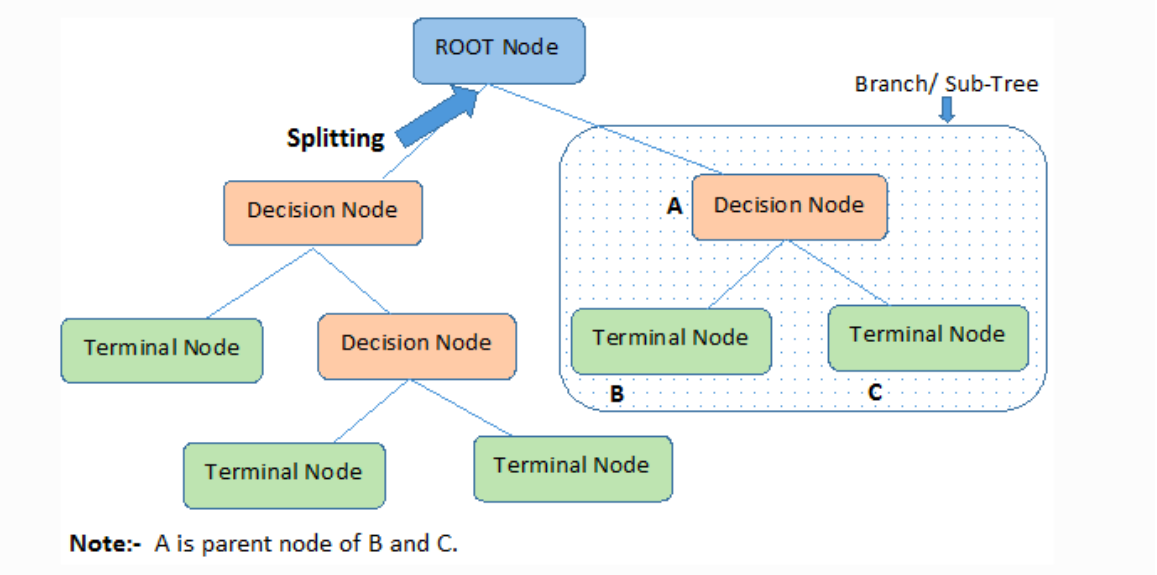


Figure 3.2.3: A schematic of a decision tree [Cha].

A random forest model, as shown in Fig. 3.2.4, is made up of these many small decision trees (known as estimators), which each give their own predictions. The random forest model then combines the predictions of the individual estimators to give a more accurate prediction. Decision tree classifiers have the disadvantage that they can easily overfit on the training data. Instead, the random forest ensemble method allows it to generalize well to the test data, including data with missing values. Random forests are also good at handling large datasets with high dimensionality [Has+09].

2. **Gradient Boosting Machine (GBM)** [Fri01] is a type of machine learning boosting algorithm. It relies on the intuition that the next model in the ensemble, when combined with previous models, minimizes the overall prediction error. Thus, the target labels for the next model are set to minimize the error

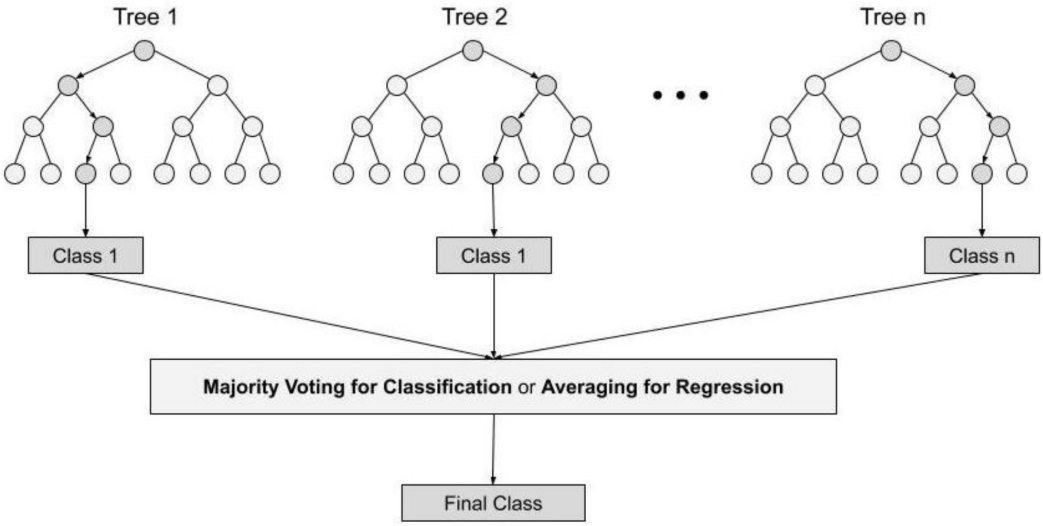


Figure 3.2.4: A schematic of a random forest.

than the previous model. Predictions from the new model are close to the targets and will reduce the error. The name gradient boosting arises because target outcomes for each case are set based on the gradient of the error with respect to the prediction. Each new model takes a step in the direction that minimizes the prediction error, in the space of possible predictions for each training case.

Gradient boosting involves three elements:

a) A *loss function* to be optimized, which depends on the type of problem being solved and must be differentiable. Many standard loss functions are supported by existing libraries, but they can also be defined by the user. For example, a regression problem may use a squared error and a classification problem may use logarithmic loss.

b) A *weak learner* to make predictions, i.e. a weak hypothesis whose performance is at least slightly better than random chance. The idea is to use the weak learning method several times to get a succession of hypotheses, each one refocused on the examples that the previous ones found difficult and misclassified [Val13]. Decision trees are used as the weak learner in gradient boosting.

c) An *additive model* to add weak learners to minimize the loss function. Trees are added one at a time, and existing trees in the model are not changed. A gradient descent optimization procedure [Rud16] is used to minimize the loss when adding trees. Traditionally, gradient descent is used to minimize a set of parameters, such as the coefficients in a regression equation or weights in a neural network. After calculating error or loss, the weights are updated to minimize that error. To perform the gradient descent procedure after calculating the loss, one adds a tree to the model that reduces the

loss (i.e. the gradient is followed). This is done by parameterizing the tree, then modifying the parameters of the tree and moving in the right direction by reducing the loss. Generally, this approach is called functional gradient descent or gradient descent with functions. The output of the new tree is then added to the output of the existing sequence of trees to improve the final output of the model. A fixed number of trees are added or training stops once loss reaches an acceptable level or no longer improves on an external validation dataset.

3. **Extreme Gradient Boosting (XGB)** [CG16] provides parallel tree boosting (Fig. 3.2.5) and is the leading machine learning algorithm for regression, classification, and ranking problems. XGB is a scalable and highly accurate implementation of gradient boosting that pushes the limits of computing power for boosted tree algorithms, being built largely for energizing machine learning model performance and computational speed. With XGB, trees are built in parallel, instead of sequentially like GBM. It follows a level-wise strategy, scanning across gradient values and using these partial sums to evaluate the quality of splits at every possible split in the training set. It is a perfect combination of software and hardware optimization techniques to yield superior results using less computing resources in the shortest amount of time.

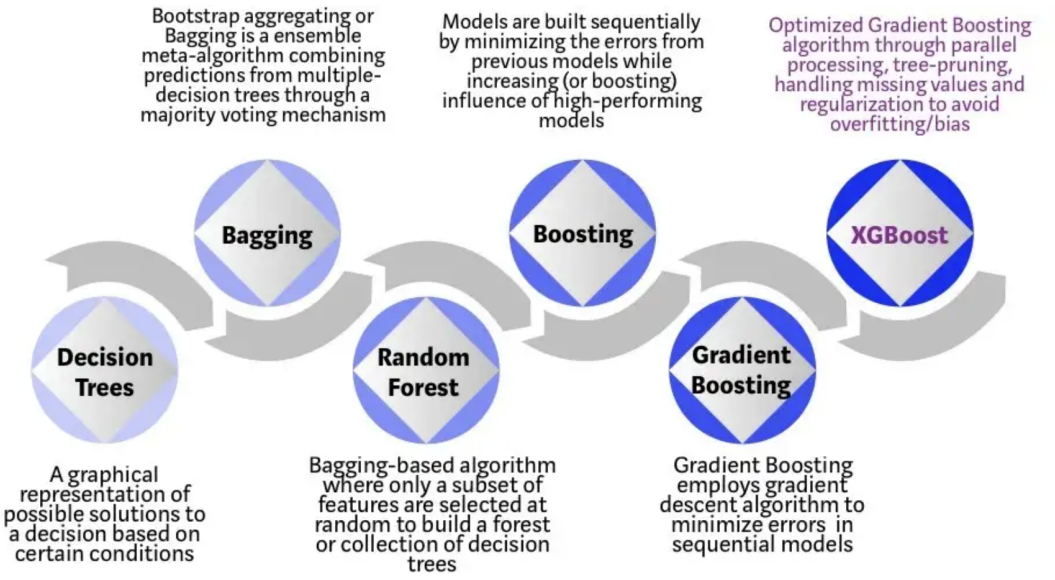


Figure 3.2.5: Evolution of XGB Algorithm from Decision Trees [MS].

3.2.5 Performance Metrics

The performance of classification predictive modeling algorithms are evaluated based on certain statistical metrics. These performance metrics are used during all three

phases of learning and testing, i.e., training, validation and testing. Some important terms to note before understanding the performance metrics are: True Positives, TP, False Positives, FP, True Negatives, TN, and False Negatives, FN The following explanation is assuming a binary classification or a two class classification problem, but it can be extended to a multi-class problem using one-vs-all predictions. A true positive is an outcome where the model correctly predicts the sample is from a positive class. Similarly, a true negative is an outcome where the model correctly predicts the sample is from the negative class. A false positive is an outcome where the model incorrectly predicts the sample to be from the positive class, and a false negative is an outcome where the model incorrectly predicts the negative class. Using TP, FP, TN, and FN a number of performance metrics can be defined [Bur19]:

1. **Accuracy**: Classification accuracy is defined as the number of correct predictions divided by the total number of predictions,

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}. \tag{3.2.6}$$

   Accuracy is useful when the dataset is well-balanced but is not a good metric choice for unbalanced datasets. When the dataset is imbalanced, the number of samples in one class is much larger than the number of samples in the other classes, accuracy cannot be considered a reliable measure anymore, because it provides an overoptimistic estimation of the classifier ability on the majority class.

2. **Sensitivity**: Sensitivity or Recall explains, out of the total number of positive samples, how many positive samples was the classifier able to predict correctly. It is a useful metric in cases where false negative is of higher concern than false positive.

$$Sensitivity = \frac{TP}{TP + FN}. \tag{3.2.7}$$

3. **Specificity**: Specificity explains out of the total number of negative samples, the number of samples that were predicted as negative by the classifier. It can be written as

$$Specificity = \frac{TN}{TN + FP}. \tag{3.2.8}$$

4. **Precision**: Precision explains, out of the total predicted positive samples, how many samples were actually positive. Precision is useful in the cases where false positive is a higher concern than false negative.

$$Precision = \frac{TP}{TP + FP}. \tag{3.2.9}$$

5. **F1-Score**: For learning from imbalanced datasets, one needs to improve recall, which as stated above provides information about a classifier's performance with

respect to false negatives, without hurting precision, which deals with false positives. Unfortunately, trying to reach this goal can often be challenging, since when increasing the true positives for the minority class, the number of false positives can also increase, reducing precision. The problem is greatly alleviated by using a metric that combines the trade-offs of both precision and recall, namely the F1- score [CJ20]. F1-score is the harmonic mean of precision and recall. The best value of F1 would be 1 (perfect precision and recall) and worst would be 0.

$$F1 - Score = \frac{2TP}{2TP + FP + FN}. \tag{3.2.10}$$

6. **Matthews Correlation Constant (MCC)**: Another metric for working with imbalanced classes is Matthews Correlation Constant (MCC) [CJ20]. It ranges in the interval 1 to +1, with extreme values –1 and +1 attained in case of perfect misclassification and perfect classification, respectively, while MCC=0 means that the classifier is no better than a random flip of a fair coin. MCC takes into account all four values TP, FP, TN, and FN and a high value (close to 1) means that both classes are predicted well, even if one class is disproportionately under or over-represented.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{3.2.11}$$

3.2.6 Evaluation

Model evaluation or testing is done at two specific points of the learning procedure, one during validation (Section 3.2.3) and the second during testing. This evaluation is performed on the basis of one or more performance metrics defined in Section 3.2.5. After learning the hyperparameters and tuning of the model is complete, one retrains the model with the selected hyperparameters and tests it on the test dataset, where the model should be able to predict the labels of the dataset. On the basis of the criteria defined using the performance metrics, the performance of the model on the test set decides whether the learnt model was satisfactory or a new model that is more suitable for the given data should be learnt or the performance metrics should be changed, or the task is complex and requires more data.

This chapter discussed the basic steps involved for a machine learning based classification workflow. Data acquisition and preprocessing is discussed. The importance of the training, validation and test datasets was elaborated. A number of basic classifier models and the performance metrics that help to estimate the performance of the learnt model were discussed. Details on when to use which performance metrics are also provided. In the next chapter, I give a detailed explanation of my method to predict hot spot residues. The next chapter also contains all the computational details of my method that can help to reproduce the method.

4 RBHS (Robust Principal Component Analysis-(RPCA) based Prediction of Protein-Protein Interaction Hot spots)

The ability of machine learning techniques to use data to learn without the need for external programming (Chapter 3), combined with an increase in data availability and more powerful hardware and software resources, has led to an increase in the popularity of machine learning (ML) algorithms in every field, including the prediction of the hot spot residues for protein-protein interactions.

The machine learning (ML) based hot spot prediction methods mentioned in the Introduction (Chapter 1) use a data matrix that contains protein sequence- and structure-based features. Such data matrices often contain values that can be corrupted by errors or noise that are caused due to experimental mistakes, computational issues and/or human blunders [Mor+17; Can+11; KC21]. As expected, this adversely affects the predictive power of the machine learning algorithms. The existing machine learning based hot spot prediction algorithms do not take this issue into account. However, machine learning algorithms are sensitive to noise in the data [GG19]. Noise in the data can result in poor prediction results and greatly decreased classification accuracy. Consequently, it becomes important to deal with the issue of noise in the data before applying any machine learning algorithms to the data. Hence, it is imperative to use an approach for ML based hot spot prediction where the data matrix contains reduced noise (an ideal scenario would be noiseless data matrix).

In my method, namely RBHS (**R**obust Principal Component Analysis-(RPCA) **B**ased Prediction of Protein-Protein Interaction **H**ot **S**pots) [Sit+21], I address this issue of noisy data matrix by pre-processing it using Robust Principal Component Analysis (RPCA) [Can+11]. RPCA is a variant of Principal Component Analysis (PCA) [WEG87; Jol02] method (Section 3.2.2.4). RPCA works by decomposing a noisy matrix D into two components, namely, a low rank component, A, and a sparse component, S, irrespective of the number of corrupted entries (i.e., robustly).

The low rank matrix A obtained after applying RPCA to the original data matrix is then the new data matrix for the pipeline of identification of hot spots. In this chapter, I talk about RBHS and the machine learning methods I used along with RBHS to predict hot spot residues. In particular, information regarding databases,

datasets, features, the proposed method RBHS is discussed in detail. To address the issue of noise in protein-protein interaction hot spot data, I propose the RBHS workflow in detail along with all its computational details.

4.1 Databases

A prerequisite for using machine learning (ML) based methods for any problem is that these methods require well-curated datasets, as talked about in detail in Chapter 3. Experimental hot spots have been cataloged in many databases. In general, for machine learning based hot spot prediction approaches, researchers combine data from multiple such databases, filter the redundancies and split the data into training, validation and test sets.

I derived the data matrices for my method as in reference [LLD18]. The authors of [LLD18] created two datasets, HB-34 and BID-18 from existing protein databases namely Alanine Scanning Energetics database (ASEdb) [TB01], the Binding Interface Database (BID) [Fis+03], Structural Kinetic and Energetic Database of Mutant Protein Interactions (SKEMPI) [MF12] and Protein–protein Interactions Thermodynamic Database (PINT) [KG06].

4.1.1 ASedb

Alanine Scanning Energetics database (ASEdb) [TB01] is one of the earliest protein interface databases. ASEdb is a searchable relational database containing current Alanine scanning data that can be updated as new data is published. ASEdb contains Alanine-scanning mutational analyses of interfaces for which changes in binding energy have been measured. In addition, solvent accessible surface areas (ASA) have been calculated for each mutant side chain. The program NACCESS [HT93] was used to calculate ASA for ASEdb calculation with default parameters. ASA has been explained in detail in Section 4.3.2. In cases where the protein-protein complex structure was available, surface areas were calculated both for the separated proteins and for the complex.

4.1.2 BID

Along with ASEdb, BID [Fis+03] is also one of the earliest available PPI database. BID uses a data mining approach for searching literature for detailed information about residues involved in protein interfaces and organizing the information into a user-friendly database. This database is continuously augmented by mining of the scientific literature for protein interaction descriptions, as well as wild-type and mutational binding energies. Based on a relational database model, the BID has six tables: Protein Info, Protein Pair, Residue a, Residue b, Interact and Reference. The information about a protein is referenced against its GenBank [Ben+12] gene identification number, which synchronizes the information in the BID with GenBank using

the gene ID as a common key. The Protein Pair table describes the interaction. The two residue tables, Residue a and Residue b, include information about the residues involved in binding. The Interact table stores the type and strength of the protein-protein interactions. The Reference table contains the source reference and the ID to its PubMed citation as a link to the primary data.

4.1.3 SKEMPI

A more exhaustive database for protein interface hot spots is Structural Kinetic and Energetic Database of Mutant Protein Interactions (SKEMPI) [MF12]. It contains the changes in the binding energies, entropy, enthalpy, and rate constants upon mutation. SKEMPI contains the most diverse data about experimental hotspots that was first presented in 2012 [MF12] and later updated in 2019 to SKEMPI 2.0 [Jan+19].

4.1.4 PINT

Protein–protein Interactions Thermodynamic Database (PINT) [KG06] contains the data of several thermodynamic parameters along with sequence and structural information about protein-protein interactions, experimental conditions and literature information. Each entry contains numerical data for the free energy change, dissociation constant, association constant, enthalpy change, heat capacity change etc. of the interacting proteins upon binding. This data is important for understanding the mechanism of protein–protein interactions. PINT also includes the name and source of the proteins involved in binding, their SWISS-PROT [Boe+03] and Protein Data Bank (PDB) [Ber+00b] codes, secondary structure, and solvent accessibility of residues at mutant positions, experimental conditions, such as buffers, ions and additives, measuring methods, and other literature information.

4.2 Datasets

HB-34 and BID-18 were created by the authors of reference [LLD18] from existing protein databases namely, Alanine Scanning Energetics database (ASEdb) [TB01], the Binding Interface Database (BID) [Fis+03], Structural Kinetic and Energetic Database of Mutant Protein Interactions (SKEMPI) [MF12], and Protein–protein Interactions Thermodynamic Database (PINT) [KG06].

4.2.1 HB-34 and BID-18

For the construction of HB-34 dataset, alanine-mutation data was extracted from four databases, Alanine Scanning Energetics (ASEdb) [TB01], SKEMPI database [MF12], Ab+data [Ass+10], and Alexov_sDB [PLA15]. Then, the proteins present in the BID dataset [Fis+03] were excluded, and the redundant proteins were also removed. This resulted in the creation of a benchmark of 34 protein complexes (HB-34) with 313

interface residues. The dataset HB-34, with columns containing PDB ID, residue ID and the label indicating whether the residue is hot spot (label=1) or null spot (label=0), is presented in Table 7.1.1 in the chapter supplementary information [1].

An independent test dataset, namely BID-18, was constructed by selecting complexes from the BID database that are non-homologous to the complexes in the training dataset (HB-34). BID-18 consists of 18 protein complexes and 126 interface residues. The dataset BID-18 has columns containing PDB ID, residue ID and the label indicating whether the residue is hot spot (label=1) or null spot (label=0) and is presented in Table 7.2.1 in the chapter supplementary information. The authors in reference [LLD18] mentioned that the HB-34 and BID-18 datasets are completely independent.

However, I also analyzed both the datasets using the CD-HIT-2D web server [Hua+10]. The FASTA sequences of both the datasets given as input to the web server are provided in Section 7.13 and Section 7.14 of the Supplementary information. I tried to identify sequences that were similar, keeping a stringent criterion as follows:

1. the sequence identity should be larger than 40% and
2. the coverage should be larger than 20% of the whole sequence.

Except for coagulation factor VIIA, no protein with the above characteristics could be identified. However, VIIA forms a complex with the soluble tissue factor (PDB code 1DAN) in the BID-18 dataset, and with the peptide exosite inhibitor E-76 (PDB code 1DVA) in the HB-34 dataset. It is important to note that the two protein partners are not evolutionarily related (sequence identity lower than 20% and sequence coverage below 20%). Consequently, it was expected that the results of my work will not be affected by the presence of the coagulation factor VIIA protein in common between the two datasets, HB-34 and BID-18.

As mentioned before, residues in HB-34 dataset were used to construct the training data matrix and residues in BID-18 were used to construct the test data matrix for my pipeline. It is important to note that HB-34 consists of 34 protein complexes with 313 interface residues, out of which 133 are hot spots and 180 are null spots. BID-18 consists of 18 protein complexes and 126 interface residues. Out of 126 residues in BID-18, 39 are hot spots and 87 are null spots. These are summarized in Table 4.2.1.

[1] These tables are long and hence have been included in the Supplementary information, instead of this chapter.

Dataset	No. of Protein Complexes	No. of Interface Residues	No. of Hot Spots	No. of Null Spots
HB-34	34	313	133	180
BID-18	18	126	39	87

Table 4.2.1: Details of the two datasets, HB-34 and BID-18.

The next step in the pipeline for using ML for hot spot identification is to encode various sequence and structure based features for protein-protein interaction interface residues by using existing bio-informatics tools and software.

4.3 Encoding the residues as features

Many protein structure and sequence based features or attributes are collected using a number of bioinformatics tools to represent protein-protein interaction interface residues for machine learning methods of hot spot prediction. These features are:

1. Sequence based features.
2. Structure based features.

Each of them is discussed in detail next and is also summarized in Table 4.3.1.

4.3.1 Sequence based features

Protein sequence features include physicochemical properties of amino acids, evolutionary information in terms of evolutionary conservation score and position-specific scoring matrix (PSSM) and other similar descriptors that encode the sequence information of the residues.

Hot spots are evolutionarily more conserved than other residues, and this information is incorporated as sequence based features in machine learning based hot spot prediction methods. The calculation of evolutionary conservation scores is done using multiple sequence alignments and phylogenetic trees [Ash+10]. Another commonly used sequence feature is Position-specific scoring matrices (PSSMs) that can be obtained from NCBI non-redundant databases via PSI-BLAST [Alt+97]. The features I use for my hot spot prediction method also comprise these sequence features, as explained below.

4.3.1.1 Physicochemical features

The physicochemical properties of amino acids in my dataset are calculated from the AAIndex [Kaw+07] database because it is an important resource for studying the physicochemical and biochemical properties of amino acids and their roles in

protein structure, function, and evolution. It has been invaluably used to analyze protein sequences and generate features from these sequences, that can be used to train algorithms and models for various bioinformatics applications. The database is freely available online and can be also be downloaded as a file. Each entry in the AAIndex database represents a specific index which is a particular property and consists of a unique identifier, a short description of the index, and a set of numerical values assigned to the 20 standard amino acids. The values typically range from positive to negative, representing the extent of a particular property possessed by each amino acid. Five physicochemical features (hydrophobicity, hydrophilicity, polarity, polarizability, average accessible surface area) obtained from the AAIndex database [Kaw+07] and one more physicochemical property (propensity) [JT97] have been used in my work:

1. Normalized consensus hydrophobicity is a measure used in biophysics studies to calculate the relative hydrophobicity of amino acids in a protein sequence [Mal+08]. The Eisenberg scale [Eis+84] in the AAIndex database was used in this thesis to calculate normalized consensus hydrophobicity. It is based on the principle that hydrophobic amino acids tend to be buried inside the protein's core, away from the surrounding water molecules, while hydrophilic amino acids are exposed to the solvent. The scale is typically normalized to a reference amino acid, which is assigned a value of zero, and other amino acids are ranked based on their relative hydrophobicity compared to the reference [Mal+08; Eis+84]. These values are derived from experimental data and/or theoretical calculations, taking into account various factors such as solvent accessibility, side chain interactions, and protein folding.

2. Hopp-Woods hydrophilicity scale [HW81], is a scale that estimates the hydrophilicity of amino acids based on their propensity or tendency to be located on the surface of proteins and exposed to water. This scale was originally developed by William Hopp and Harold Woods in 1981 to predict regions of proteins that are likely to be antigenic. However, it is also used in bioinformatics to check for hydrophilicity of amino acids in the protein sequence. The Hopp-Woods scale assigns numerical values to each amino acid based on their hydrophilic properties, with positive values indicating hydrophilic amino acids and negative values indicating hydrophobic amino acids.

3. Polarity is calculated using the Grantham polarity index [Gra74], which is a widely used index in the AAIndex database to quantify the polarity of amino acids. This index is based on the idea that certain amino acids have polar or charged side chains, while others have nonpolar side chains, and this index assigns numerical values to the 20 amino acids based on their relative polarity. The values are derived from the differences in chemical properties, like charge distribution and dipole moment, between the amino acids. In, this index, higher values indicate greater polarity.

4. Polarizability parameter developed by [CC82] is used to calculate the polarizability of amino acids in my work, and it represents the average polarizability of the side chains for each amino acid. Higher values indicate greater polarizability of the side chain, implying a higher susceptibility to polarizable interactions. This index specifically focuses on the polarizability of amino acid side chains, rather than the polarizability of the entire amino acid or individual atoms within the side chains. This index from the AAIndex database used to quantify the polarizability characteristics of amino acid side chains is quite useful in understanding intermolecular interactions, ligand binding, protein-protein interactions, and other aspects of protein structure and function [CC82].

5. The accessibility to the solvent is measured from the accessible surface area of each residue in the protein complex [Jan+78]. The authors in [Jan+78] calculated an average accessible surface area scale according to which residues are classed as buried if their average accessible surface area is smaller than 20 Angstrom, exposed if it is larger than 60 Angstrom and intermediate if A is between 20 and 60 Angstrom. This scale is present in AAIndex database, and it was also used as a physicochemical property in my pipeline to predict hot spot interface residues.

6. Residue interface propensity is used to understand which amino acid residues have a higher tendency to be present at the interfaces as compared to other amino acids in a protein complex. According to [JT96] the residues with higher tendency to be present at the interface were often involved in important protein interactions, such as hydrogen bonding, hydrophobic contacts, or salt bridges. In my work I used the natural logarithmic of residue interface propensities as mentioned in [JT97] and the higher the logarithmic propensity, the more likely a residue is to occur in a protein-protein interface.

These features are reported in Table 7.5.1 for residues in HB-34 and Table 7.6.1 for residues in BID-18 [1] .

4.3.1.2 Position-Specific Score Matrix based features

Twenty position-specific score matrix (PSSM) profiles, calculated using PSI-BLAST [Alt+97]. Position-Specific Score Matrix (PSSM) based features are used in bioinformatics to represent the evolutionary and conservation information of residues in protein sequences. PSSMs are constructed from multiple sequence alignments of homologous sequences in the BLAST database [Alt+97], and the score for each position in the position specific scoring matrix indicates how likely it is for an amino acid residue to be present at that position based on evolutionary conservation, where positive scores indicate high likelihood, negative scores indicate low frequency. PSSM-based features are useful for distinguishing between conserved and variable regions in proteins. These

[1] These tables are long and hence have been included in the Supplementary information instead of this chapter.

features are reported in Table 7.7.1 for residues in HB-34 and Table 7.8.1 for residues in BID-18.

4.3.1.3 Block substitution matrix based features

Twenty block substitution matrix based features, computed using Blosum62 [HH92]. Blosum62 is a substitution matrix used for sequence alignment to calculate the relative frequencies of amino acids and their substitution probabilities, and 62 indicates that 62 % level of sequence similarity is taken into account. A block substitution matrix is generated by comparing a set of aligned homologous sequences, and the matrix captures substitution patterns across the entire sequence alignment. For each residue pair in the matrix, a score is assigned based on their observed substitution frequency. Positive scores indicate a high probability of substitution, while negative scores indicate low substitution likelihood. These features are reported in Table 7.3.1 for residues in HB-34 and Table 7.4.1 for residues in BID-18.

4.3.2 Structure based features

As explained in Section 2.1.1.3, the tertiary structure of a protein is the folding arrangement of the amino acids in three dimensions. The structural information is helpful to understand the function of proteins and the effect of disease associated mutations at the molecular level. Now, with the release of AlphaFold- Multimer [Eva+21], the ease of finding *in silico* structures for protein complexes has significantly increased. Thus, the available structural information of proteins complexes in the form of structural features can be incorporated into the data matrix for better hot spot prediction rather than using only sequence-based features. The structure based features I used for RBHS are:

4.3.2.1 Solvent accessible area features

One commonly used structural feature of residues is the solvent Accessible Surface Area (ASA) that is defined as the locus of the center of the virtual solvent molecule as it rolls over the surface of the protein [LLD18]. Five solvent accessible area features [RS94a] computed using Dictionary of Protein Secondary Structure (DSSP) [Joo+10] were used in this thesis. $MASA$ that is the solvent accessible surface area for a monomer, and $CASA$ that is the solvent accessible area for the protein complex are calculated first, and then the following values are calculated on them:

$$\Delta ASA = MASA - CASA \tag{4.3.1}$$

$$SB_r = \frac{ASA}{MASA} \tag{4.3.2}$$

Here, SB_r is the relative surface burial.

$$RASA = \frac{MASA}{\text{Nominal Maximum Area}} \tag{4.3.3}$$

Here, $RASA$ is relative accessible surface area and Nominal maximum area values have been mentioned in [RS94b] for all the amino acids.

$$\Delta RASA = (RASA \text{ in molecule}) - (RASA \text{ in complex}) \tag{4.3.4}$$

Now, $MASA$, ΔASA, SB_r, $RASA$ and $\Delta RASA$ values are used as features in my work. These features are reported in Table 7.9.1 for residues in HB-34 and Table 7.10.1 for residues in BID-18.

4.3.2.2 Solvent Exposure features

Seven solvent exposure features, computed using hsexpo [Ham05] are used in this thesis. From [Ham05], Half-sphere exposure (HSE) is an excellent measure of solvent exposure of an amino acid residue in a complex and HSE separates a residue sphere into two half-spheres: HSE-up corresponds to the upper sphere in the direction of the chain side of the residue, while HSE-down points to the lower sphere in the direction of the opposite side [Son+08]. Now, seven exposure features are calculated using the software hsexpo [Ham05] are:

1. $HSEAU$: the number of C_α atoms in the upper sphere.
2. $HSEAD$: number of C_α atoms in the lower sphere.
3. $HSEBU$: the number of C_β atoms in the upper sphere.
4. $HSEBD$: the number of C_β atoms in the lower half sphere.
5. Coordination number (CN): the number of C_α atoms within a sphere around the C_α atom of a residue.
6. Residue depth (RD): atom depth is defined as the distance between a given atom and the nearest point on the solvent accessible surface. Residue depth is the average atom depth of a residue's atoms.
7. RD_α: the atom depth of a residue's C_α atom.

The analysis of solvent exposure features is useful for understanding protein structure and function, as solvent-exposed residues are more likely to be involved in interactions with other molecules, such as ligand binding, protein-protein interactions, or interactions with water molecules. These features are reported in Table 7.11.1 for residues in HB-34 and Table 7.12.1 for residues in BID-18. These tables are long and hence have been included in the Supplementary information instead of this chapter.

After combining all the structure and sequence based features described above, there is a total of 58 features in the data matrix. The resulting training data matrix is of size 58×313 because there are 58 features and 313 residues in HB-34. The testing data matrix is of size 58×126 because there are 126 residues in BID-18 dataset.

Feature Characteristic	**No. of features**	**Feature names**	**Tool/Database**
Sequence based features			
Physicochemical	6	Hydrophobicity, hydrophilicity, polarity, polarizibility, propensities and average accessible surface area	AAIndex database [Kaw+07], propensities using [JT97; JT96]
Position-Specific Score Matrix (PSSM)	20	Sequence alignment scores with respect to 20 target frequencies for each position in the query protein.	PSI-BLAST [Alt+97]
Block substitution matrix	20	Alignment score matrix after comparison of sequences with pairwise identity no more than 62%.	Blosum62 [HH92]
Structure based features			
Solvent Accessible Surface Area(ASA)	5	ASA (monomer), *Delta* ASA, Relative ASA (RASA), *Delta* RASA	DSSP [Joo+10]
Solvent exposure features	7	HSEBD, HSEAU, HSEAD, HSEBU, CN, RD, and RDa	hsexpo [Ham05]

Table 4.3.1: The sequence and structure based features used in both the datasets. A total of 58 features are used, which includes 46 sequence based features and 12 structure based features.

4.4 Workflow

Hot spot prediction is done using these data matrices as inputs to the machine learning classifiers. For better understanding, I have described the steps of the method in a workflow. Fig. 4.4.1 illustrates the proposed workflow, and it involves the following steps,

Step 1: Pre-processing the data using RBHS.

Step 2: Training and validation of suitable machine learning models.

Step 3: Applying the models on test data and predicting the output labels.

In the subsequent sections, each of these steps are described in detail.

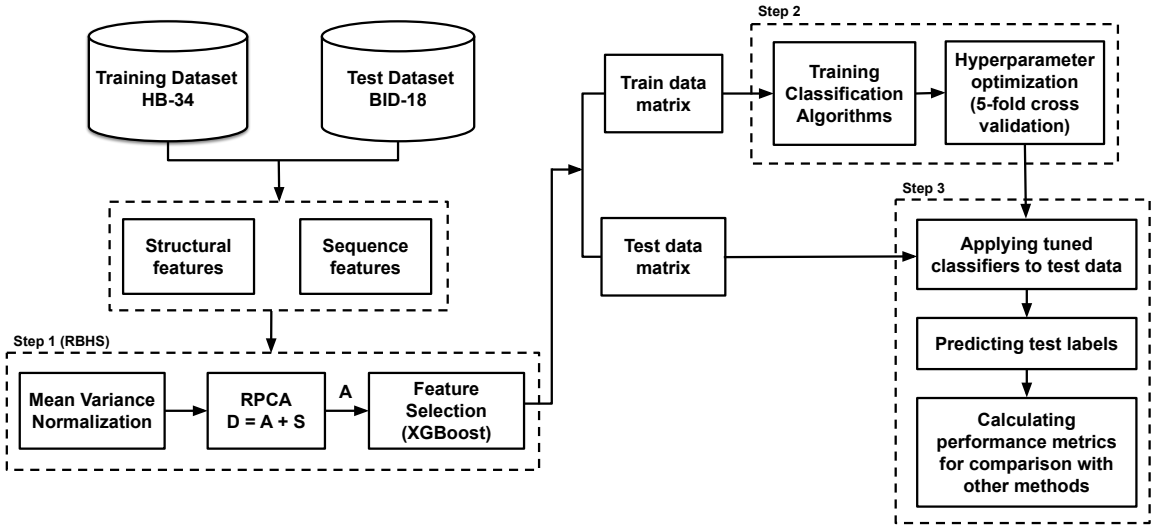


Figure 4.4.1: Workflow illustrating the steps of my approach for hot spot prediction.

4.4.1 Step 1: RBHS

RBHS is a novel pre-processing pipeline for recovering a data matrix with reduced noise from a noisy data matrix and then performing feature selection on the reduced noise matrix. The RBHS pipeline for data pre-processing is composed of the following steps:

1. **Normalization**: In the first step, the data matrices (mathematical notation: D) that have been constructed from the training dataset (HB-34) and from the testing dataset (BID-18) respectively (called training data matrix and test data matrix for the rest of this thesis) are normalized. Normalization is usually done on a data matrix because in most cases the values of different features in the data matrix have different scales. This results in inaccurate predictions made by the classifier. In my data matrices (described in Sections 4.2 and 4.3)

it can be seen from Tables 7.3.1, 7.4.1, 7.5.1, 7.6.1, 7.7.1, 7.8.1, 7.9.1, 7.10.1, 7.11.1, 7.12.1 that different features have different numerical ranges or scales. For example, the maximum and minimum value of the Blosum62 features lie between −4 and 11 and are integers, for both the datasets, HB-34 and BID-18, but the hydrophilicity feature, for both datasets, has its maximum and minimum value as −3.4 and 3 respectively and are represented by real numbers. Thus, it becomes important to normalize features to the same scale before further processing. I tested various normalization techniques, as explained in Scikit-Learn [Ped+c]. Finally, I used mean variance normalization (Section 3.2.2) on the training and test data matrices because it gave the best results among the different techniques that I tested.

2. **RPCA**: Next, I apply Robust Principal Component Analysis (RPCA) to both normalized training and normalized test data matrices and recover the corresponding matrices A that contain reduced noise. Biological data often contains noise, which may lead to inaccuracies in the prediction of the machine learning classifiers. I used RPCA for pre-processing the data because it splits the data matrix D, which may have corrupted entries, into a low-rank matrix A containing reduced noise and a sparse matrix S, ($D = A + S$) [Can+11]. Consider a matrix of size $m \times n$ with $m < n$. Here, m is the number of rows and n is the number of columns in the matrix. Then the matrix is a full rank matrix when all m rows are linearly independent. Now, a set of vectors is called linearly independent if no vector in the set can be expressed as a linear combination of the other vectors in the set and consequently the rank of the matrix will be m. On the other hand, when $m > n$, the matrix is full rank when all n columns are linearly independent and the rank of the matrix is the number of columns that is n. The matrix that does not have full rank is a low rank matrix [Str06]. In other words, all columns or rows are not linearly independent. A sparse matrix is a matrix in which most of the entries or elements are zero [Str06]. In my thesis, the non-zero elements of the sparse matrix correspond to the putative corrupted or noisy entries. There are several mathematical approaches to solve RPCA [BZ14] in literature. In my method, I use the Principal Component Pursuit (PCP) method as described in [Can+11]. My codes to solve RPCA is in [Sit23].

   To calculate A and S, the following optimization problem is formulated:

$$\begin{aligned} &min_{A,S} \|A\|_* + \lambda \|S\|_1 \\ &\text{subject to } D = A + S \end{aligned} \tag{4.4.1}$$

   Here, $\|.\|_*$ is the nuclear norm of the matrix A. The nuclear norm of a matrix is the sum of the singular values of the matrix. The l_1-norm of the matrix S, written as $\|S\|_1$, is the sum of the absolute values of the entries of the matrix. λ is a regularization parameter. The value of λ was determined experimentally to obtain the best values of performance metrics (Section 3.2.5).The details to

solve Equation 4.4.1 can be found in reference [Ara+14] and Section 4.5.1.1. The authors in [Can+11] show mathematically that the low rank matrix A that is recovered from D using RCPA contains reduced noise and the matrix S is sparse. Moreover, without loss of any information, S can be assumed to contain noise and can be discarded. The results of applying RPCA to the data matrix D of the training set HB-34 has been shown in Fig. 5.1.1 in Section 5.1, and it can be seen from the figure that D contains entries corrupted by noise, and these appear as random, spike-like elements in the matrix. A is the matrix with reduced noise obtained from D after applying RPCA to it. S is the sparse matrix that contains noise and can be safely discarded without loss of information. Finally, A can be used as the new data matrix for both training and test sets.

3. **Feature Selection**: Feature selection has been explained in detail in Section 2 of Chapter 3. Briefly, feature selection is the process of selecting the most important features to input to machine learning algorithms. It is done to reduce the number of input variables by eliminating redundant or correlated features and thus, the data matrix now contains the most relevant set of features as input to machine learning models.

   I performed feature selection on training and test matrices A, to obtain reduced training and test matrices A'. To perform feature selection, feature importance for all the features can be calculated using various techniques. In my work, the feature importance of all the features in the training and test matrices A were calculated using the SciKit-learn library in reference [Ped+b] and the Extreme Gradient Boosting (XGB) algorithm [CG16]. I selected those features in both training and test A matrices, whose feature importance is above an empirically calculated threshold. The details to calculate this threshold are provided in Section 4.5.1.2 of this chapter. Feature selection helps to identify the effective feature subspace for building the prediction models, and for RBHS two new data matrices A' are obtained. The matrix, A' that is a reduced matrix, has 51 features, instead of the original 58 for both training and test data.

4.4.2 Step 2: Training and validation

The next step is the training and validation of classification algorithms on the training data set matrix A' obtained from RBHS. I trained several popular machine learning classifiers including Support Vector Machines (SVM) [Guy+02; BGV92], Gradient Boosting Machines (GBM) [Fri01], Extreme Gradient Boosting (XGB) [CG16], and Random Forests [Bre01] on the training data matrix A'. Details about these classifiers are in Section 3.2.4.

Next, I used 5-fold cross validation for hyperparameter tuning of the trained classifiers using Scikit-learn [Ped+11a; Ped+a]. More information about validation is provided in Section 3.2.3. In 5-fold cross validation, the training data is divided into five subsets. One of the five subsets is used as the validation

set, and the other four are combined to form a training set. This method is repeated five times. Hence, every data sample is in the validation set exactly once and in the training set four times. During the 5-fold cross validation, I used F1-score as the scoring parameter to assess the efficacy of the classifier model. The F1-Score calculated in each of the five times is averaged over all 5 iterations. Cross-validation is extremely beneficial because not only does it help to find hyperparameters for classification algorithms, but it also helps to reduce overfitting on the training set. This is because the dataset is split into multiple folds or subsets, and the algorithm is trained each time on a different fold. This helps to make the model more generalizable to any dataset.

4.4.3 Step 3: Testing

I applied the validated models on A' to calculate labels for the interface residues in the test set, BID-18. As explained before, hot spot prediction is a binary classification problem where if the label=1, the residue is classified as a hot spot and if the label=0, it is a null spot. The computationally predicted hot and null spot labels are compared with the experimentally annotated labels and several performance metrics are calculated. A detailed description of each of these metrics and why and when they are used is described in Section 3.2.5. It is important to note that TP are the number of true positives which means that the predicted hot spot residues are also known experimentally to be hot spots, FP are the false positives which means that predicted hot spot residues are experimentally identified as null spots, TN are the true negatives i.e., the predicted null spots are experimental null spots as well and FN are the false negatives, which means predicted null spots are experimental hot spots. For brevity, the performance metrics are again defined here as follows:

$$Recall = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

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

$$Precision = \frac{TP}{TP + FP}$$

$$F1 - Score = \frac{2TP}{2TP + FP + FN}$$

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

Here, MCC is the Mathew's, Correlation Coefficient. In the next chapter, various methods are compared using these performance metrics.

4.5 Computational details

In this section, the computational details of RBHS and the computational details of the machine learning algorithms used with RBHS have been described. The proposed method RBHS was coded in MATLAB and Jupyter notebook. All the experiments were performed on MATLAB R2019a and Jupyter Notebook (Python 3.6) running on a MacBook Pro with the following specifications: MacBookPro (13-inch, 2017, two Thunderbolt 3 Ports) Processor: 2.3 GHz Dual-Core Intel Core i5 Memory: 8 GB 2133 MHz LPDDR3 Hard disk: 512 GB SSD

The main libraries I imported in Jupyter notebooks are: Scikit-Learn [Ped+11a], numpy [Har+20], pandas [McK+10], matlab.engine, xgboost [CG16], and matplotlib [Hun07]. I used Scikit-learn [Ped+11a] because it is a comprehensive library for machine learning in Python. It has efficient tools for machine learning and statistical modeling that help to perform tasks like classification, regression, clustering, and dimensionality reduction using a consistent interface in Python. I ran my Matlab codes for RPCA from my jupyter notebooks (Python) with the help of the MATLAB Engine API for Python. It allows integrating MATLAB functionality directly with a Python application, creating an interface to run a MATLAB file from Python code.

NumPy [Har+20] is a library for Python that provides support for large, multi-dimensional arrays and matrices and a huge collection of mathematical functions to operate on these arrays. Pandas is a software library written for the programming language Python to perform data manipulation and analysis. I employed Extreme gradient boosting by using XGBoost [CG16] that is an open-source software library that provides a regularizing gradient boosting framework for various languages like C++, Java, Python, R, Julia, Perl, and Scala. It works on various operating systems like Linux, Windows, and macOS. I used Matplotlib [Hun07] to make plots, graphs and for data visualization.

Various parameters for different steps in the workflow (shown in Fig. 4.4.1), that I used to predict hot spots, are mentioned below.

4.5.1 RBHS

4.5.1.1 RPCA

As mentioned before, RPCA can be solved by a technique called Principal Component Pursuit(PCP). PCP has two variables that need to be calculated, A and S. Here I use the method of alternating minimization, i.e., solve for A keeping S constant and then

solve for S keeping A constant. This alternation is done for a number of iterations till the desired accuracy in the estimates of A and S is reached. At each iteration k, the solution for S is the soft-thresholding algorithm [Sel] and A is the singular value thresholding algorithm also known as nuclear norm minimization [CCS10; Can+11]. Calculation of A is the more computation intensive part, as it requires calculation of the Singular Value Decomposition (SVD) [Str19] at each iteration. The decomposition performance of the RPCA is evaluated on the basis of the final classification accuracy of an SVM. RPCA algorithm required two hyperparameters: the first is λ as can be seen from Equation 4.4.1 and the second is the number of iterations k. For each of these hyperparameters, a range of values was given to the algorithm and the value of *lambda* and k at which an SVM model achieved the highest value of evaluation metrics like accuracy, F1-score etc. during training was chosen. For λ, a wide range of values from 1×10^{-6}, 1×10^{-5}, ..., 0.1, 0.2,...,0.8, 0.9, 1×10^{5}, 1×10^{6} was used. The values of number of iterations that were used for hyperparameter tuning were: 10, 20, 30, 40, 50, 100, 200. Finally, the best value of λ turned out to be 0.3 and the best value of the number of iterations was 30.

Hyperparameter	Final Value
λ	0.3
iterations, k	30

Table 4.5.1: Values of the hyperparameters used for Robust Principal Component Analysis (RPCA). For further details on the method, I refer the reader to section 4.4.1.

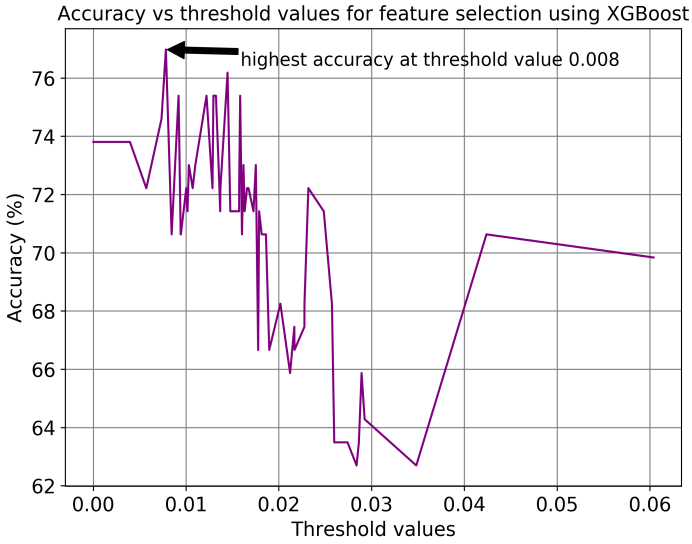
4.5.1.2 Feature Selection

As mentioned in 4.4 to perform feature selection on the low rank matrix A obtained from RPCA, I select those features in both training and test A matrices, whose feature importance is above an empirically calculated threshold. To determine this threshold, I observed whether the performance of the XGB classifier in terms of its accuracy increases or decreases with the number of selected features. Then the value of feature importance, at which the maximum value of accuracy is observed, is set as the threshold for feature selection. All the features that are above this threshold are kept, and those that have feature importance below this threshold are discarded. The plot of accuracy values *versus* the threshold values can be seen in Fig. 4.5.1.

After observing the plot, it can be concluded that the optimal value of the threshold is 0.008. The accuracy-threshold curve is rough, and other threshold values may also lead to accurate results as well.

4.5.2 Training and Validation

I perform 5-fold cross validation values of hyperparameters for each classification algorithm:



(a)

Figure 4.5.1: Accuracy *vs* threshold values plot for feature selection using Extreme Gradient Boosting (XGB). It can be seen that the highest value of accuracy of the XGB classifier is at the threshold value 0.008. All features with feature importance less than 0.008 are discarded from the data matrix, and only those features are selected whose feature importance is more than or equal to 0.08.

4.5.2.1 SVM

For support vector machines (SVM) [Guy+02; BGV92] I experimented with four different SVM kernels, namely, linear, sigmoid, polynomial, and radial basis function (rbf). SVM separates the data into different categories by finding the best hyperplane and maximizing the distance between points (Section 3.2.4.1). But when the data is not linearly separable, a kernel function is used. The kernel function is just a mathematical function that converts a low-dimensional input space into a higher-dimensional space. This is done by mapping the data into a new feature space. In this space, the data will be linearly separable and SVM can find the hyperplane that separates the data into different classes. Some of the commonly used kernel functions are the linear kernel, the polynomial kernel, the RBF kernel, and the sigmoid kernel. The range of values that I used for hyperparameters for the kernels are tabulated in Table 4.5.2. It is important to note that if no range of a particular hyperparameter will be passed in the grid search, then the default value of that hyperparameter as specified by Scikit-learn [Ped+11a] is used by the model.

Kernel	**gamma**	C	**degree**
Polynomial	1×10^{-5}, 1×10^{-4}, 1×10^{-3}, 1×10^{-2}, 1×10^{-1}, 1	1×10^{-3}, 1×10^{-2}, 1×10^{-1}, 1, 1×10^{1}, 25, 50, 1×10^{2}, 500, 1×10^{3}, 1500, 2000	2 - 15
RBF	1×10^{-5}, 1×10^{-4}, 1×10^{-3}, 1×10^{-2}, 1×10^{-1}, 1	1×10^{-3}, 1×10^{-2}, 1×10^{-1}, 1, 1×10^{1}, 25, 50, 1×10^{2}, 500, 1×10^{3}, 1500, 2000	-
Sigmoid	1×10^{-4}, 1×10^{-3}, 1×10^{-2}, 1×10^{-1}, 1	1×10^{-3}, 1×10^{-2}, 1×10^{-1}, 1, 1×10^{1}, 25, 50, 1×10^{2}, 500, 1×10^{3}, 1500, 2000	-
Linear	-	1×10^{-3}, 1×10^{-2}, 1×10^{-1}, 1, 1×10^{1}, 25, 50, 1×10^{2}, 500, 1×10^{3}, 1500, 2000	-

Table 4.5.2: Hyperparameter ranges for hyperparameter tuning using grid search for different SVM kernels.

After hyperparameter tuning, the best performing SVM model had a polynomial kernel with a degree of 3, gamma of 0.01 and C of 500 (Table 4.5.3).

Hyperparameter	Final Value
Kernel	Polynomial
degree	3
gamma	0.01
C	500

Table 4.5.3: Values of the hyperparameters obtained from grid search using 5-fold cross validation for a support vector machine (SVM) classifier. The best results during 5-fold cross validation were obtained when SVM was used with a polynomial kernel of degree=3, with the value of C=500 and gamma=0.01. It is important to note that if no range of a particular hyperparameter will be passed in the grid search, then the default value of that hyperparameter as specified by Scikit-learn [Ped+11a] is used by the model.

4.5.2.2 GBM

Boosting is a sequential technique which works on the principle of ensembles. It combines a set of weak learners and delivers improved prediction accuracy. At each particular iteration, a new weak learner model is trained with respect to the error of the whole ensemble learnt so far. For more details, please refer to Section 3.2.4.2. A thorough grid search for best values of hyperparameters for GBM classifier was performed using 5-fold cross validation on the training data. The performance metric that was used to quantify which combination of hyperparameters gives the best results is the negative Mean Squared Error (negMSE). The Mean Squared Error (MSE) takes the difference between the actual values and those predicted by the model and find the mean of the squares. negMSE will return a negated version of the calculated MSE. The hyperparameters of a GBM [Fri01] used in Scikit-learn for cross validation are explained below.

1. min_samples_split: It defines the minimum number of samples which are required in a node to be considered for splitting. It is used to control overfitting of the GBM model. Lower values can result in overfitting of the model to the data, and too high values can lead to underfitting. The range of values I passed to a GBM classifier for cross validation are : 2, 3, 4.

2. min_samples_leaf: It defines the minimum samples required in a terminal node or leaf. It is also used to control the overfitting of the model on the training data. The range of values I passed to a GBM classifier for 5-fold cross validation are : 1, 2, 3, ..., 57, 58, 59.

3. max_depth: This is the maximum depth of a tree and is used to control overfitting, in the sense that a higher depth will allow the model to learn relations very specific to a particular sample. The range of values I passed to the GBM classifier : 1, 2, 3, ...7, 8, 9, 10.

4. max_features: This is the number of features to consider while searching for the best split. As a thumb-rule, the square root of the total number of features works great, but one should check up to 30-40% of the total number of features. Higher values can lead to overfitting. The range of values for this hyperparameter that I passed to the model are: 1, 2, 3, 4, 5, ..., 12.

Hyperparameter	**Values**
min_samples_split	2, 3, 4
min_samples_leaf	1, 2, ..., 58, 59
max_depth	1, 2, ..., 9, 10
max_features	1, 2, ..., 11, 12
n_estimators	10, 20, 30, ..., 1980, 1990, 2000
learning rate	0.2
subsample	0.6, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95

Table 4.5.4: Hyperparameter ranges for hyperparameter tuning using grid search for GBM.

5. n_estimators:The number of sequential trees to be modeled. Though GBM is fairly robust at higher number of trees, it can still overfit. Hence, it needs to be fine-tuned. The range of values used for grid search were: 10, 20, 30, 40, ..., 1980, 1990, 2000.

6. learning rate: This determines the impact of each tree on the final outcome. GBM works by starting with an initial estimate, which is updated using the output of each tree. The learning parameter controls the magnitude of this change in the estimates. Lower values are generally preferred as they make the model robust to the specific characteristics of the tree and thus allowing it to generalize properly. However, lower values would require higher number of trees to model all the relations and the process will be computationally expensive. So, I chose the value of learning rate as 0.2 because it was low but not very low to make my algorithm computationally expensive.

7. subsample: The fraction of observations to be selected for each tree. Selection is done by random sampling. The values of subsample rate passed to grid search were: 0.6, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95.

The range of values for each of the hyperparameter used for tuning are summarized in Table 4.5.4. After the grid search, the best value of hyperparameters obtained for GBM classifier are tabulated in Table 4.5.5.

Hyperparameter	Final Value
min_samples_split	2
min_samples_leaf	31
max_depth	1
max_features	7
n_estimators	1500
subsample	0.9

Table 4.5.5: Values of the hyperparameters obtained from grid search using 5-fold cross validation for Gradient Boosting Machine (GBM) classifier.

4.5.2.3 Extreme Gradient Boosting

Extreme Gradient Boosting (XGB) [CG16] (Section 3.2.4.2) is a scalable and advanced implementation of gradient boosting that pushes the limits of computing power for boosted tree algorithms, being built largely for improving machine learning model's performance and computational speed. With XGB, trees are built in parallel, instead of sequentially like GBM. After a thorough grid search using 5-fold cross validation on the training data, the best values of hyperparameters for XGB classifier were calculated and these hyperparameters of XGB are explained as follows:

Hyperparameter	**Values**
min_child_weight	1, 3, 4, 5, 6, 8, 10, 12
max_depth	3, 4, 5, 6, 9
gamma	0.1, 0.2, 0.3, 0.4, 0.5
subsample	0.6, 0.7, 0.75, 0.8, 0.85, 0.9, 1
colsample_bytree	0.6, 0.7, 0.75, 0.8, 0.85, 0.9, 1

Table 4.5.6: Hyperparameter ranges for hyperparameter tuning using grid search for XGB.

1. min_child_weight: Defines the minimum sum of weights of all observations required in a child. This is similar to min_child_leaf in GBM, but not exactly. This refers to the minimum sum of weights of observations, while GBM has the minimum number of observations. This hyperparameter is used to control overfitting. Higher values prevent a model from learning relations that might be highly specific to the particular sample selected for a tree. But very high values

can lead to underfitting; hence, it should be tuned using cross validation. The values of this hyperparameter I used for grid search are: 1, 3, 4, 5, 6, 8,10,12.

2. max_depth: The maximum depth of a tree in XGB is the same as GBM, and it is used to control overfitting, as higher depth will allow the model to learn relations very specific to a particular sample. The values of this hyperparameter I used for grid search are: 3, 4, 5, 6, 9.

3. gamma: A node is split only when the resulting split gives a positive decrease in the loss function. Gamma specifies the minimum loss decrease required to make a split. The values of this hyperparameter I used for grid search are: 0, 0.1, 0.2, 0.3, 0.4, 0.5.

4. subsample: It is the same as the subsample of a GBM. It denotes the fraction of observations to be used as random samples for each tree. Lower values prevent overfitting, but very small values can lead to underfitting. The values of this hyperparameter I used for grid search are: 0.6, 0.7, 0.75, 0.8, 0.85, 0.9, 1.

5. colsample_bytree: This is similar to the hyperparameter max_features used in GBM. It denotes the fraction of features that will be used to train each tree. The values of this hyperparameter I used for grid search are: 0.6, 0.7, 0.75, 0.8, 0.85, 0.9, 1.

The search ranges for each hyperparameter of the XGB classifier is summarized in Table 4.5.6. After the grid search, the best value of hyperparameters obtained for XGB classifier are tabulated in Table 4.5.7.

Hyperparameter	Final Value
min_child_weight	1
max_depth	3
gamma	0
subsample	1
colsample_bytree	1

Table 4.5.7: Values of the hyperparameters obtained from grid search using 5-fold cross validation for Extreme Gradient Boost (XGB) classifier.

4.5.2.4 Random Forest

Random Forest (RF) [Bre01] is an ensemble algorithm that combines multiple decision trees to make predictions for a classification problem. I performed a thorough grid search using cross validation to find out the best set of hyperparameters for the random forest model. The various random Forest hyperparameters I fine-tuned are:

Hyperparameter	Values
min_depth	3, 5, 8, 13, 20, 25, 30, 35, 40
min_sample_split	2, 3, ..., 11, 12
min_samples_leaf	2, 3, 4, 5, 10, 15, 20, 26
n_estimators	100, 150, 200, 250, 300, 350
max_features	2, 7, 14

Table 4.5.8: Hyperparameter ranges for hyperparameter tuning using grid search for Random Forest.

1. max_depth: The max_depth of a tree in a Random Forest is defined as the longest path between the root node and the leaf node. A higher value can result in overfitting of the model. The range of values I passed for max_depth are: 3, 5, 8, 13, 20, 25, 30, 35, 40.

2. min_sample_split: This is a hyperparameter that tells the decision tree in a random forest the minimum required number of observations in any given node in order to split it. By increasing the value of the min_sample_split, one can reduce the number of splits that happen in the decision tree and therefore prevent the model from overfitting. The range of values I used for this hyperparameter are: 2, 3, 4, 5, ..., 7, 10, 11, 12.

3. min_samples_leaf: This Random Forest hyperparameter specifies the minimum number of samples that should be present in the leaf node after splitting a node. This hyperparameter also helps prevent overfitting as its value increases. The range of values I used for this hyperparameter are: 2, 3, 4, 5, 10, 15, 20, 26.

4. n_estimators: A Random Forest algorithm is a grouping of trees. The hyperparameter n_estimators is the number of decision trees considered. More trees should be able to produce a more generalized result, but by choosing more trees, the time complexity of the Random Forest model also increases. Hence, an optimum value of this hyperparameter should be considered. I passed the following range of values of this hyperparameter: 100, 150, 200, 250, 300, 350 to RF model for grid search.

5. max_features: This is the number of maximum features provided to each tree in a random forest. It is a good convention to consider the default value of this parameter, which is set to the square root of the number of features present in the dataset. The ideal number of max_features generally tend to lie close to this value. The range of values I used for this hyperparameter are: 2, 7, 14.

The range of hyperparameters used during hypertuning is summarized in Table 4.5.8. After performing grid search using 5-fold cross validation, the best hyperparameters for Random Forest classifier are provided in Table 4.5.9. I performed a grid search with the range of values of the hyperparameters I mentioned above for each of the

Hyperparameter	Final Value
max_depth	13
min_sample_split	4
min_samples_leaf	3
n_estimators	200
max_features	14

Table 4.5.9: Values of the hyperparameters obtained from grid search using 5-fold cross validation for Random Forest (RF) classifier.

classifiers I used for my work, using 5-fold cross validation on the training data matrix. I used the best values of hyperparameters for each classifier and with the help of these validated classifiers, I predicted hot spots on the independent test set. An important point to note here is that, if the hyperparameter values are within a certain limit of the best values obtained with the help of cross-validation, the results will not get affected drastically. In fact, the prediction results will still be fairly close to the results calculated with the best values of classifier hyperparameters.

Another important point to note is that if no range of a particular hyperparameter will be passed during grid search, then the default value of that hyperparameter as mentioned in Scikit-learn [Ped+11a] is used by the model.

I have provided all the data that was used for my method to predict hot spots in the Supplementary Information (Chapter 7). The data tables could not be included in this chapter because they were lengthy. The codes for the implementation of my method for hot spot prediction can be found at [Sit23].

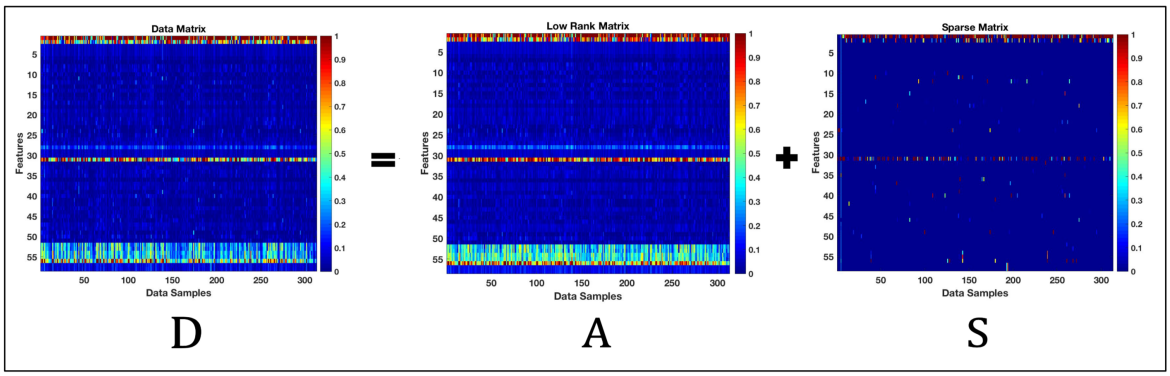
The next chapter compares the results of using RBHS along with the classifiers described earlier with PCA and original data. It also compares the performance of various classifiers with each other when used with RBHS, to see which classifier performs best for the task of hot spot prediction. Finally, the comparison of my method with the state-of-the-art methods for hot spot prediction is also shown in the next chapter.

5 Results

In the previous chapter, the algorithmic and computational details of my method RBHS, to predict hot spots have been described. In this chapter, to show the efficacy of RBHS, preliminary, intermediate and final results obtained from using RBHS along with various classifiers will be presented in detail.

5.1 RPCA

A workflow depicting the steps of my approach to predict hot spots can be seen in (Fig. 4.4.1) in Section 4.4. In step 1 of the workflow (Fig. 4.4.1), the original noisy data matrix, D, that contains the sequence and structure based features of interface residues, is first normalized. Then it is decomposed using Robust Principal Component Analysis (RPCA) [Can+11] to obtain a noise-reduced low rank matrix A. The results of applying RPCA to the residues in HB-34 training set is shown in Fig. 5.1.1.



(a)

Figure 5.1.1: RPCA is applied to the data matrix D of the training set HB-34. D contains entries corrupted by noise, and these appear as random, spike-like elements in the matrix. A is the matrix with reduced noise obtained from D after applying RPCA to D. S is the sparse matrix that contains noise and can be safely discarded without loss of information.

From Fig. 5.1.1, it can be seen that the original data matrix D is corrupted by noise, which can be seen from the figure as random, spike-like elements in D. This

noise is inherent to the data matrix and is caused by human and/or computational errors in the process of calculating structure and sequence based features for various residues in the dataset. The low rank matrix A that is recovered from D using RCPA is seen to have reduced noise. Moreover, it can be seen that the matrix S is sparse. Therefore, S can be assumed to contain noise and can be discarded.

Hence, it can be assumed that the data matrices obtained from the HB-34 [LLD18] and BID-18 [LLD18] datasets after applying RPCA do in fact contain reduced noise.

5.2 Comparing RBHS with PCA and Original Data

The first step to see the effectiveness of my pre-processing pipeline RBHS to predict hot spots is to see how well RBHS (applied to the original data) works when compared to the feature extraction algorithm, Principal Component Analysis (PCA) (applied to the same data) and when compared to the original data without any pre-processing.Thus, three matrices are used:

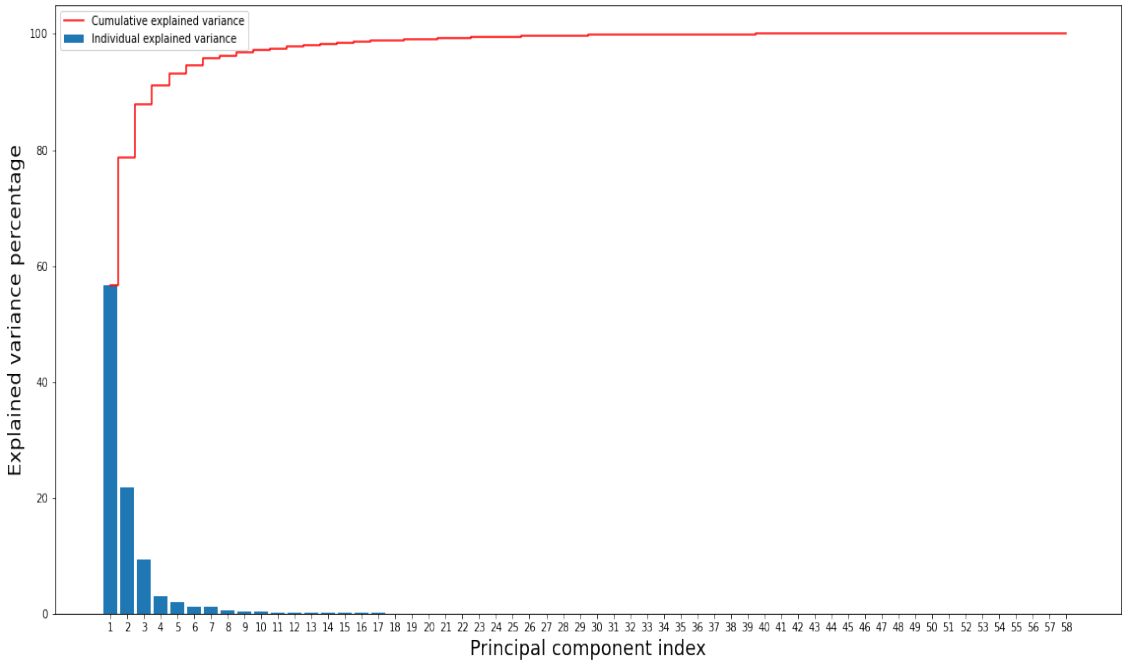
1. The original data matrix D. This data matrix contains the original features calculated on the residues of HB-34 and BID-18 without any pre-processing done on it. The details of these features are in Section 4.3.

2. The matrix that has been obtained after applying Principal Component Analysis (PCA) to the data matrix D.

3. The reduced matrix A' (calculated in Section 4.4) that is obtained after performing my approach RBHS [Sit+21] on the original data matrix D.

5.2.1 PCA applied to HB-34

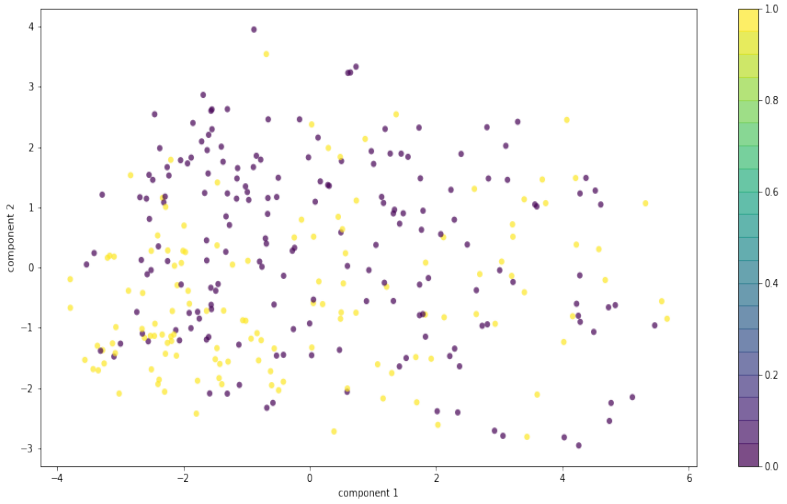
PCA was implemented using the Scikit-learn library [Ped+d]. For the representation of PCA, the principal components explaining 95% variance were chosen. For doing this, a plot showing different percentage of explained variances versus the principal components on the training data set matrix was made, as shown in Fig. 5.2.1. The step plot represents the cumulative variance explained by a particular number of principal components on the x-axis. It can be seen from the plot, after the 7 principal components, the explained variance becomes constant. Thus, the first 7 principal components can be selected because they explain 95% variance of the data. The rest of the components can be safely discarded because they do not explain significant variance in the data. The resulting data matrix contained 7 features after applying PCA to the original data matrix [WEG87; Jol02].



(a)

Figure 5.2.1: The x-axis represents the principal component index. The y-axis represents the explained variance percentage. Each bar indicates how much variance a particular component captures itself. The step plot represents the cumulative variance explained by a particular number of principal components on the x-axis.

Next, I plot the first two principal components of residues in HB-34 dataset to learn more about the data. As described in Section 3.2.2.4, one of PCA's main drawback is that it tends to be highly affected by outliers in the data. The assumption when using PCA is that the principal components, with the highest variance, will be the most useful in predicting if a residue belongs to one of two classes, namely, hot spots and null spots. The principal component with the highest variance would be the best feature that would allow separating the residues into hot spots and null spots. As can be seen from Fig. 5.2.2, PCA fails to separate the two classes. Hence, PCA does not provide meaningful class information when there is noise in the data, as is the case with the data used in this thesis.



(a)

Figure 5.2.2: The first two principal components after the PCA transformation of original data matrix D of residues in HB-34. The data points denoted in purple belong to the null spot class, and data points in yellow are the hot spots.

Next, I show the performance of various classifiers mentioned in Section 4.4.2, during training and validation on the dataset HB-34. This is Step 2 of the workflow (Fig. 4.4.1).

5.2.2 Performance of classifiers on the training data

Classification algorithms including Support Vector Machines (SVM) [Guy+02; BGV92], Gradient Boosting Machines (GBM) [Fri01], Extreme Gradient Boosting (XGB) [CG16] and Random Forests (RF) [Bre01] are now trained and validated on all the three training data matrices. This includes the original data matrix D that contains structure and sequence based features of residues in HB-34, the data matrix obtained after applying PCA to D, and the matrix obtained after applying RBHS to D. To analyze the performance of these classifiers, performance metrics that include recall (Equation 3.2.7), specificity (Equation 3.2.8), accuracy (Equation 3.2.6), precision (Equation 3.2.9), Mathew's Correlation Constant (MCC) (Equation 3.2.10), and the F1-score (Equation 3.2.11) as described in Section 4.4.3 are used.

It is important to note that both the datasets used in this thesis, HB-34 [LLD18] (Table 7.1.1) and BID-18 [LLD18] (Table 7.2.1), have imbalanced classes. Imbalanced classes in hot spot prediction means, that the number of hot spots are fewer than the number of null spots. HB-34 has 133 hot spots and 180 null spots, and BID-18 has 39 hot spots and 87 null spots. This imbalance will cause the classification algorithm to

Method	Recall	Specificity	Accuracy	Precision	F1-Score	MCC
Original Data+ SVM	0.59	0.80	0.71	0.69	0.63	0.40
PCA+SVM	0.36	0.89	0.67	0.72	0.48	0.30
RBHS+SVM	0.67	0.74	0.71	0.66	0.66	0.41
Original Data+GBM	0.65	0.78	0.72	0.69	0.66	0.43
PCA+GBM	0.58	0.71	0.65	0.60	0.59	0.29
RBHS+GBM	0.67	0.73	0.71	0.65	0.66	0.40
Original Data+XGB	0.57	0.74	0.67	0.62	0.59	0.32
PCA+XGB	0.61	0.71	0.67	0.61	0.60	0.31
RBHS+XGB	0.56	0.76	0.68	0.64	0.59	0.33
Original Data+RF	0.51	0.79	0.67	0.65	0.57	0.31
PCA+RF	0.53	0.74	0.67	0.57	0.52	0.24
RBHS+RF	0.56	0.78	0.68	0.65	0.60	0.34

Table 5.2.1: Performance comparison of various methods on the training dataset HB-34. These values are computed in Step 2 of our workflow in Fig. 4.4.1.

be biased towards predicting a residue as a null spot because there are more null spots than hot spots. To solve this problem of imbalanced classes, the metric F1-score is considered for the analysis and interpretation of the results shown in the tables below. A detailed description on which performance metric is useful in which case is provided in Section 3.2.5. From that discussion, it can be followed that for learning from imbalanced datasets one needs to improve recall/sensitivity (Equation 3.2.7), which provides information about a classifier's performance with respect to false negatives, without hurting the precision (Equation 3.2.9), which deals with false positives.

For practical purposes, achieving this goal can be quite challenging. This is because when increasing the true positives for the minority class (in this case the class of hot spots), the number of false positives can also increase, which will result in a reduced value of precision (Chapter 3). F1-score is a metric that combines the trade-offs of both precision and recall because it is the harmonic mean of precision and recall. Its highest value is 1 (when there is perfect precision and recall) and lowest value is 0. Consequently, to deal with the issue of imbalanced classes, F1-score was used as the scoring parameter for tuning the hyperparameters during training and validation steps (Section 4.5.2) and also to assess the performance of my method for hot spot prediction.

Table 5.2.1 shows the performance of the popular machine learning classifiers as mentioned earlier upon implementing 5-fold cross validation on the training dataset HB-34. Using RBHS with SVM and GBM classifiers and Original Data with GBM, gives the highest value of F1-score (0.66) while PCA+SVM and PCA+RF gives the lowest value i.e., 0.48 and 0.52, respectively.

5.2.3 Performance of classifiers on the test data

Next, the trained models were applied to the testing data BID-18 to predict test class labels (Table 5.2.2). As can be observed, the best value of F1-score on the test set is obtained by RBHS+XGB (0.66) and RBHS+SVM (0.64), while the lowest value is attained by Original Data+GBM (0.47) and PCA+GBM (0.46). So, RBHS performs better than PCA and Original data, regardless of the classifier being used.

Method	Recall	Specificity	Accuracy	Precision	F1-Score	MCC
Original Data+ SVM	0.79	0.66	0.70	0.51	0.62	0.42
PCA+SVM	0.59	0.67	0.71	0.44	0.51	0.24
RBHS+SVM	0.80	0.69	0.72	0.53	0.64	0.45
Original Data+GBM	0.54	0.66	0.62	0.41	0.47	0.18
PCA+GBM	0.54	0.64	0.61	0.40	0.46	0.17
RBHS+GBM	0.69	0.76	0.74	0.56	0.62	0.43
Original Data+XGB	0.54	0.78	0.71	0.53	0.53	0.32
PCA+XGB	0.59	0.67	0.64	0.44	0.51	0.24
RBHS+XGB	0.72	0.79	0.77	0.61	0.66	0.49
Original Data+RF	0.54	0.80	0.72	0.55	0.54	0.34
PCA+RF	0.56	0.76	0.70	0.51	0.54	0.31
RBHS+RF	0.67	0.78	0.75	0.58	0.62	0.43

Table 5.2.2: Performance of different methods on the test dataset BID-18. These values are computed in Step 3 of our workflow in Fig. 4.4.1.

Now, the results of training are compared with the results of testing. There is a significant increase in F1-scores of RBHS+XGB and RBHS+RF during testing whereas, for RBHS+SVM, the F1-score does not change and gets slightly decreased for RBHS+GBM in Table 5.2.2. Hence, it can be concluded that there is no overfitting on the training data when using RBHS. On the other hand, F1-scores of classifiers applied to the original data during testing (Table 5.2.2) are overall lower than F1-scores of classifiers applied to the original data during the training (Table 1). Mostly, this is caused by the overfitting of the classifiers on the original training data during training. A similar observation can be made from Tables 5.2.1 and 5.2.2 for PCA, except for PCA+SVM, where there is a slight increase of 0.03 during testing. It can be seen from Fig. 5.1.1 that the matrix D contains a significant amount of noise, and thus the PCA algorithm generates a noisy representation of D. As mentioned before, PCA is sensitive to noise in the data, and classifiers that are trained on these noisy representations usually do not perform well. Similarly, classifiers using data representations based on the original matrix D tend to overfit on the noisy training data and, thus, give poor predictions on the test data [Can+11]. In contrast, in my method, I use the matrix A' that contains reduced noise and that has been obtained from the noisy matrix D using RBHS (Section 4.4). Therefore, my model does not overfit on the training data and works well during both training and testing.

5.2.4 Receiver Operating Characteristic and Precision-Recall Curve

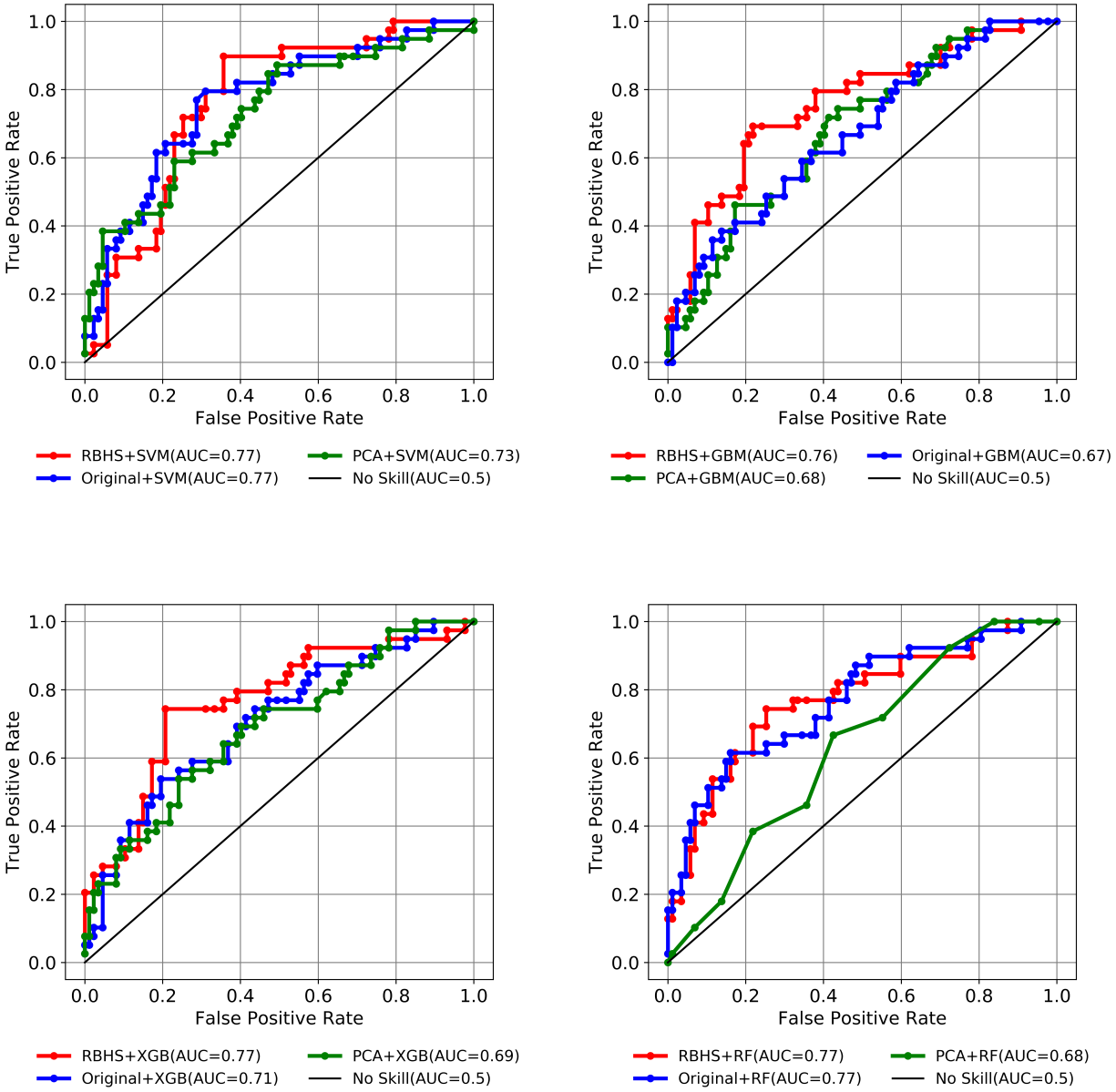


Figure 5.2.3: ROC (Receiver Operating Characteristic) Curves to compare the performance of all the methods on the independent test set BID-18 along with the AUC (Area under the curve) values for each method.

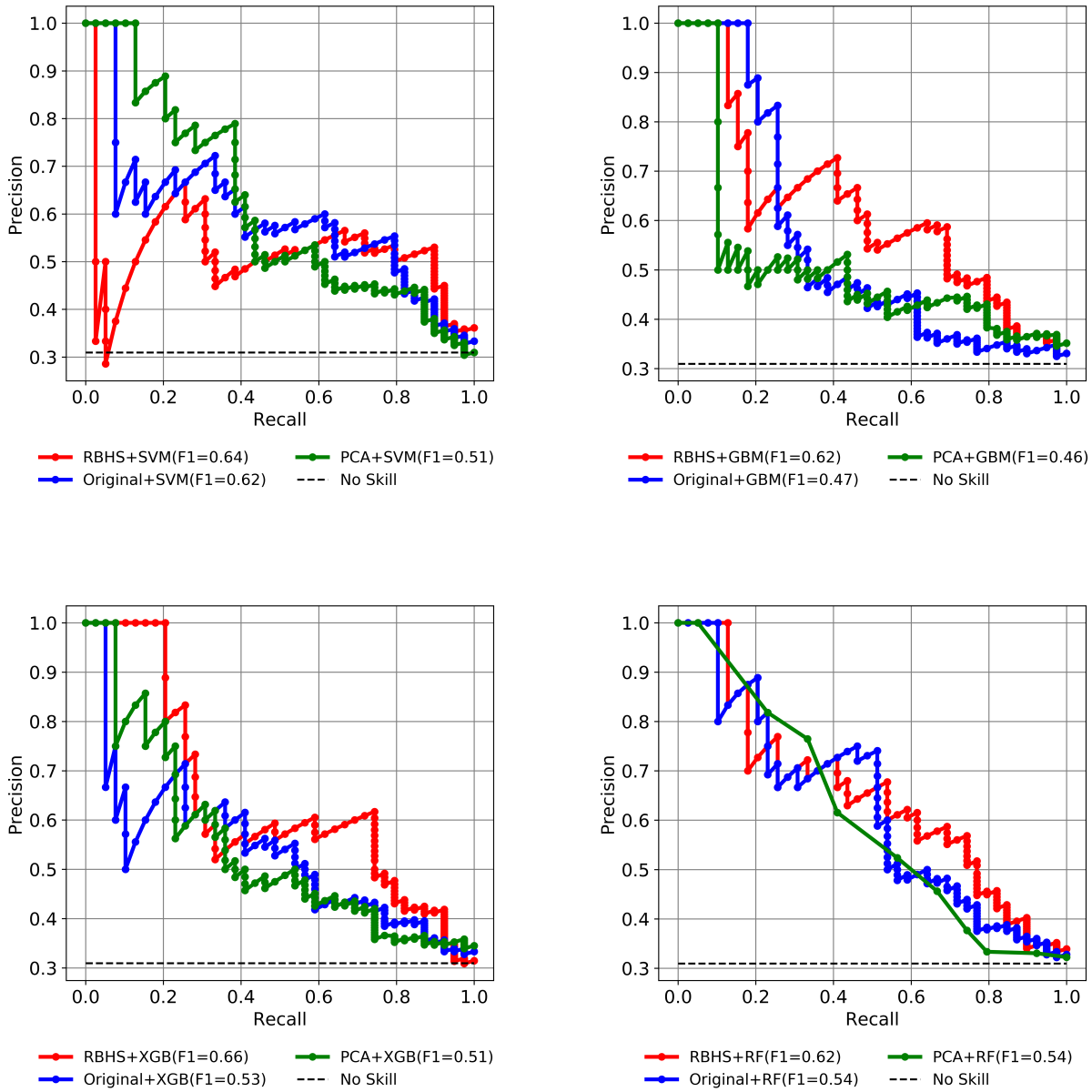


Figure 5.2.4: Precision-Recall Curves of different methods applied on the independent test set. The F1-Score values for each method are also reported.

Next, I calculated the receiver operating characteristic (ROC) curve along with the Area Under the ROC curve (AUROC), Fig. 5.2.3, because this measures the discriminative power of the algorithm, which in this case is the ability of the algorithm to correctly differentiate hot spots residues from null spots. I also plot the Precision-Recall curves, Fig. 5.2.4, because they are more informative in case of imbalanced classes in the datasets [SR15]. The next step is to find the best classifier from all the classifiers

I used with RBHS during training and testing, with the help of Table 5.2.2, the ROC curves (Fig. 5.2.3), and from the Precision-Recall curves (Fig. 5.2.4). We can see from Table 5.2.2 and the precision-recall curves that RBHS+SVM applied to the residues in the test set BID-18 performs best for recall with a value of, 0.8 and RBHS+XGB performs best for precision and achieves a precision value of 0.61. RBHS+XGB also performs best for other metrics like F1-score (0.66), accuracy (0.77), and MCC (0.49). It performs second best for specificity (0.79), after Original Data+RF (0.80). Hence, it can be concluded that RBHS when used along with Extreme Gradient Boosting (XGB) classifier shows a reliable performance in identifying hot spots. Therefore, for comparing my hot spot prediction approach with the other state-of-the-art methods mentioned in Chapter 1, I use Extreme Gradient Boosting (XGB) classifier.

5.3 Comparison of RBHS+XGB with state-of-the-art methods

To test the predictive power of the proposed method, the performance of RBHS+XGB is compared with other state-of-the-art hot spot prediction techniques that are HEP [Xia+16], PredHS-SVM [Den+13], KFC2a and KFC2b [ZM11], PCRPi [Ass+10; SAF10], MINERVA [CKL09a], APIS [CKL09b], KFC [DPM07], Robetta [KB02], and FOLDEF [GNS02] on the independent test set BID-18 (Table 5.3.1). More details on the these techniques have been given in Chapter 1.

Method	Recall	Specificity	Accuracy	Precision	F1-Score	MCC
PredHS-SVM	**0.79**	**0.93**	**0.83**	0.59	**0.68**	**0.57**
HEP	0.60	0.76	**0.79**	**0.84**	**0.70**	**0.56**
RBHS+XGB	**0.72**	0.79	**0.77**	0.61	**0.66**	**0.49**
KFC2a	0.55	0.73	0.73	**0.74**	0.63	0.44
KFC2b	0.64	0.87	**0.77**	0.55	0.60	0.44
MINERVA	**0.65**	**0.90**	0.76	0.44	0.52	0.38
APIS	0.57	0.76	0.75	**0.72**	0.64	0.45
Robetta	0.52	**0.88**	0.72	0.33	0.41	0.25
FOLDEF	0.48	**0.88**	0.69	0.26	0.34	0.17
PCRPi	0.51	0.75	0.69	0.39	0.44	0.25
KFC	0.48	0.85	0.69	0.31	0.38	0.19

Table 5.3.1: Comparison of proposed approach (RBHS) when used with XGB classifier (known as RBHS+XGB), with other state-of-the-art methods for hot spot prediction. For each performance metric, the top scoring method is highlighted in blue, the second one in green and the third one in yellow.

Comparing the performance of my method with the state-of-the-art methods is not an easy task because there is a lack of defined training and test sets used by

the existing approaches. Different methods are trained on different sets of residues, or their trained models are not available in most cases, or sometimes it is difficult to obtain the methods. Moreover, the website provided by the authors is no longer accessible.

Yet, to compare my method to the state-of-the-art (SOTA) methods for hot spot identification, I have used the predictions made by these SOTA methods on the dataset BID-18 as specified by [LLD18]. As can be seen from Table 5.3.1, my method (RBHS+XGB) turns out to be in the top three after HEP [KB19] and PredHS-SVM [Fri01] based on the F1-score value. Moreover, it also performs second best in case of Recall (sensitivity) values and third best in terms of Accuracy and MCC values.

F1-Score is the best indicator for the predictive power of the methods with imbalance datasets, as has been described earlier in this chapter. Hot spot datasets HB-34 and BID-18 are also imbalanced. Hence, the results in Table 5.3.1 establish unambiguously the predictive power of my method for hot spot prediction.

5.4 Results for different thresholds in feature selection

The empirically calculated threshold for feature selection in the experiments for this thesis, step 1 of the workflow in Fig. 4.4.1, is 0.008 (Section 4.5.1.2). Here, I performed additional experiments, to show that a slight variation in the threshold does not change the final results dramatically. In particular, I varied the threshold slightly and studied the effects of another threshold value, i.e. 0.009. The F1-score of using RBHS with various classifiers, decreases by 0.05 or less on using 0.009 as the threshold instead of 0.008, as can be seen from Table 5.4.1. This suggests that modifying the threshold by a small extent does not affect the results dramatically. But choosing cutoff values larger than 0.016, which result in lower accuracies, decreases significantly the performance of the RBHS+XGB method, as can be seen from Table 5.4.2.

Next, the effects of using different threshold values for feature selection (Step 1c of the workflow, Fig. 4.4.1) along with the method RBHS+XGB are reported in Table 5.4.2. It can be concluded that, for the method RBHS+XGB, varying the threshold for feature selection slightly still produces good results for the prediction of hot spots. However, any dramatic change in the threshold might result in inaccurate prediction of hot spot residues.

5.5 Adding artificial noise to the data

Although the results presented in this thesis are based on the training matrix constructed by computing 58 features from HB-34 dataset, I performed additional tests using other training matrices to see how the addition of noise affects the predictions made by RBHS+XGB. The noise is generated by artificially introducing Gaussian noise (with zero mean and either 1 or 10 standard deviation). This noise is randomly

threshold	Method	Recall	Specificity	Accuracy	Precision	F1-Score	MCC
0.08	RBHS+SVM	0.80	0.69	0.72	0.53	0.64	0.45
0.09	RBHS+SVM	0.72	0.71	0.71	0.53	0.61	0.40
0.08	RBHS+GBM	0.69	0.76	0.74	0.56	0.62	0.43
0.09	RBHS+GBM	0.62	0.79	0.65	0.57	0.59	0.40
0.08	RBHS+XGB	0.72	0.79	0.77	0.61	0.66	0.49
0.09	RBHS+XGB	0.72	0.77	0.75	0.58	0.64	0.46
0.08	RBHS+RF	0.67	0.78	0.75	0.58	0.62	0.43
0.09	RBHS+RF	0.62	0.79	0.65	0.57	0.59	0.40

Table 5.4.1: Testing the effects of different threshold values for feature selection (Step 1c of the workflow (Fig. 4.4.1) using Extreme Gradient Boosting (XGB) algorithm. The results of using RBHS with different feature selection thresholds with various classifiers (Step 2 of the workflow, Section 4.4.2) are reported here.

threshold	Recall	Specificity	Accuracy	Precision	F1-Score	MCC
0.0080	0.72	0.79	0.77	0.61	0.66	0.49
0.0090	0.72	0.77	0.75	0.58	0.64	0.46
0.0120	0.62	0.77	0.72	0.55	0.58	0.37
0.0130	0.62	0.81	0.75	0.60	0.61	0.43
0.0145	0.70	0.79	0.76	0.6	0.64	0.47
0.0158	0.72	0.77	0.75	0.58	0.64	0.46
0.0167	0.67	0.75	0.72	0.54	0.60	0.40
0.0175	0.56	0.80	0.73	0.56	0.56	0.37
0.0186	0.62	0.75	0.71	0.52	0.56	0.35
0.0200	0.62	0.71	0.68	0.49	0.55	0.31
0.0217	0.59	0.71	0.67	0.48	0.53	0.29
0.0230	0.51	0.82	0.72	0.56	0.53	0.34
0.0248	0.51	0.80	0.71	0.54	0.53	0.32
0.0259	0.49	0.70	0.63	0.42	0.45	0.18
0.0289	0.44	0.76	0.66	0.45	0.44	0.20
0.0340	0.59	0.64	0.63	0.43	0.49	0.43
0.0420	0.67	0.72	0.71	0.52	0.58	0.37
0.0600	0.59	0.75	0.70	0.51	0.55	0.33

Table 5.4.2: Testing the effects of different threshold values for feature selection (Step 1c of the workflow, Fig. 4.4.1) using Extreme Gradient Boosting (XGB). The results of using RBHS+XGB with different thresholds are reported here.

added to the data matrix, corrupting an increasing number of entries from 1 to 50% as shown in Table 5.5.1. These tests showed that RBHS+XGB is capable to predict

hotspots for 50% corrupted entries when using Gaussian noise of standard deviation of 1 and up to 20 % corrupted entries when using a standard deviation of the added Gaussian noise of 10 as shown in Table 5.5.1. It can be concluded, that RBHS+XGB performs significantly well in case of noisy data matrices as well.

Matrix No.	Std. dev. of noise	% Corrupted entries in the data	Accuracy	F1-Score
0+ -	0.59	Original Matrix	0.77	0.66
1	1	1	0.74	0.58
2	1	5	0.71	0.56
3	1	10	0.68	0.56
4	1	20	0.67	0.55
5	1	50	0.63	0.53
6	10	1	0.72	0.58
7	10	5	0.68	0.57
8	10	10	0.54	0.55
9	10	20	0.47	0.51
10	10	50	0.66	0.045

Table 5.5.1: Testing the effects of using different training data matrices with artificial Gaussian noise (with zero mean and standard deviation either 1 or 10) randomly added over a different number of entries. The results of using RBHS+XGB with corrupted values ranging from 1 to 50 % are reported here.

5.6 Discussion

In this work, I showed that Robust Principal Component Analysis (RPCA) was able to decompose the original noisy data matrix (containing protein sequence and structure based features calculated for interface residues) into a less noisy low rank matrix and a sparse matrix (Fig. 5.1.1). With my pre-processing pipeline, RBHS, I obtained a data matrix with reduced features. The original data matrix consisted of 58 features, while the reduced matrix had 13 features. This reduced matrix is used to train and validate various popular machine learning classifiers using 5-fold cross validation.

Using these validated classifiers on the test data, I compared the performance of my pipeline RBHS with the performance of popular pre-processing algorithm Principal Component Analysis (PCA) and also with original data that has not been pre-processed. From Tables 5.2.1 and 5.2.2, I am able to show that RBHS works better for pre-processing the data than PCA. Moreover, in Tables 5.2.1 and 5.2.2, I show that when there is no pre-processing done on the data, the classifiers perform less efficiently than using RBHS on the data. Thus, RBHS is a good pre-processing pipeline for hot spot data.

Finally, Table 5.2.2 and Fig. 5.2.3 and Fig. 5.2.4 show that RBHS when used along with Extreme Gradient Boosting (XGB) classifier gives the best results for various

performance metrics. Thus, I use RBHS+XGB for comparing my method with the state-of-the-art methods for machine learning based hot spot prediction. The values of the performance metrics in Table 5.3.1 unambiguously indicate that my method, RBHS+XGB, works well for identifying protein-protein interaction hot spots when compared to state-of-the-art methods for hot spot prediction. Moreover, the other methods use the noisy data for training and validating their models and the testing is also done on noisy data. They do not take care of this noise that is inherent to the data. Hence, predictions made on noisy data have limited accuracy. In my work, presented in this thesis, I perform RPCA on the data to reduce noise.

To further show that my method works well on noisy data matrices, I have artificially added noise to the data, as can be seen in Table 5.5.1. RBHS+XGB, still performs fairly well in predicting hot spots because RPCA in the RBHS pipeline is able to reduce noise in the data before performing any training and testing on the data.

The accuracy-threshold curve in Fig. 4.5.1 is rough, and thus it is important to see if threshold values, other than 0.008, that has been used for feature selection in this thesis, work well or not. From Table 5.4.1 and Table 5.4.2 it can be seen that if the threshold is modified by a small extent, the results will not be dramatically impacted. Thus, it can be concluded that using a threshold, slightly different from the one used in this thesis, will still yield reliable results.

6 Conclusion

Hot spots are the residues on the protein-protein interaction interface that play a crucial role in the binding free energy of the complex ($\Delta\Delta G_{binding}$) [BT98; Jan95]. The analysis of protein-protein interaction hot spots is of utmost importance from a pharmacological perspective because hot spot residues often undergo mutations in a variety of diseases [Ten+09; Pet+16a]. Moreover, with the identification of hot spots, it has become easier to target a broad range of protein-protein complexes with small molecule drugs [Pet+16b; Sco+16; Mur+17]. Thus, continual research and development in the methods available for inferring hot spots is important.

In this direction, a variety of experimental and computational approaches have been developed to identify hot spot residues. The experimental method to identify hot spot residues namely, Alanine Scanning Mutagenesis (ASM) involves the systemic point mutation of binding interface residues to Alanine, and measuring the consequent change in $\Delta G_{binding}$ ($\Delta\Delta G_{binding}$=ΔG_{mutant}− $\Delta G_{wild\text{-}type}$) [BT98]. Then if ($\Delta\Delta G_{binding}$) ≥ 2.0 kcal/mol, the interface residue is called a hot spot or else a "null spot" [MFR07; CW89; BT98]. Owing to high experimental costs and being labor-intensive, ASM is limited in the number of hot spots identified.

The computational approaches earlier used to identify hot spots were either knowledge based methods, or they used molecular dynamic simulations [GNS02; KB02; MK99; HMK02; GF08; Bre+09]. The disadvantage with these experiments is that they are not suitable for a high-throughput prediction of hot spots. At the same time, there has been an unprecedented increase in the use of Machine Learning (ML) methods to solve various biological problems pertaining to protein-protein interactions. This resulted in a prolific use of ML methods for the problem of hot spot prediction as well [Li+22; RBM22].

In computational methods, one major issue that limits accuracy of prediction, is presence of noise [Mor+17; KC21]. Such noise in the data can be ascribed to both computational and/or human errors. This can severely affect the correctness of predictions made by machine learning algorithms trained on this noisy data [GG19]. Therefore, it becomes imperative to find methods which take this issue into account. So far, the state-of-the-art machine learning based hot spot prediction methods have not addressed this issue. The machine learning algorithms they use are trained on noisy data and also the predictions are made on noisy data matrices as well. Another issue with some of the existing state-of-the-art methods [Mor+17] that use ML for predicting hot spots is that they use their training data to make final predictions. They divide their entire available data into 70% training and 30% testing. They do show intermediate test results. Eventually, instead of reporting their results on the 30% test data, they report their final results on the entire dataset available to them.

In the field of machine learning, that is inappropriate because testing a classifier on the whole or even a part of the data that it has been trained on will always give high values of performance metrics. Thus, earlier, there was a lack of clearly established training and test sets for hot spot prediction problem. In this thesis, I address both these issues. I present a novel machine learning based pre-processing technique for hot spot prediction, called **R**obust principal component analysis **B**ased prediction of protein-protein interaction **H**ot **S**pots (RBHS) and report the final results on an independent test set not used for training.

In RBHS, I recover a training and a test data matrix with reduced noise from a corrupted training and test data matrix, respectively, by using Robust Principal Component Analysis (RPCA) [Zha+15]. Next, I remove redundant features using feature selection from the less noisy data matrices. The resulting training matrix is then used to train and validate several popular machines learning classification algorithms, including Support Vector Machines (SVM), Gradient Boosting Machines (GBM), Extreme Gradient Boosting (XGB), and Random Forests (RF) (Table 5.2.1). These trained classifiers are then used to make prediction of whether the interface residue is a hot spot or a null spot on the test data matrix (Table 5.2.2). An important point to note here is that I have not used any part of the residues in the test set to train the classifiers, and they are tested on never-before-seen data for better “generalization". Generalization refers to how well a classifier is trained to classify or make predictions on unseen data.

For my thesis I used two datasets, HB-34 [LLD18] (Table 7.1.1) and BID-18 [LLD18] (Table 7.2.1), for training and testing respectively. To verify that there was no overlap between the two sets, I analyzed both the datasets using the CD-HIT-2D web server [Hua+10] and tried to identify sequences that were similar, keeping a stringent criterion as follows: the sequence identity should be larger than 40% and the coverage larger than 20% of the whole sequence. The test set BID-18 turned out to be independent to the training set HB-34. So, in my work, I used HB-34 for training and validation purposes and BID-18 as an independent test set.

From extensive experimentation, I was able to show that RBHS, when used with Extreme Gradient Boosting (XGB) classifier [CG16], gives better performance for hot spot prediction than other classifiers. I also showed that the data matrices obtained from the HB-34 [LLD18] and BID-18 [LLD18] datasets after applying RPCA did in fact contain reduced noise. My method when applied to the independent test set BID-18 was able to predict 77% of the known hot spots of the complexes investigated in Table 7.2.1 correctly, and these results can be verified from Table 5.2.2.

Comparing the performance of my method with the published state-of-the-art methods is difficult because of the following reasons:

1. There is a lack of clear training and test sets used by the methods. Different methods are trained on different sets of residues.

2. Their trained models are not available in most cases.

3. Sometimes the difficulty of obtaining the methods themselves. This is because

some databases are no longer available. In the sense that either the website provided by the authors is no longer accessible or sometimes the webpage is down.

Still, to compare my method to the state-of-the-art machine learning based methods for hot spot identification, I used the results of predictions made by these methods on the dataset BID-18 as mentioned in [LLD18]. As can be seen from Table 5.3.1, my proposed method gives good values of various performance metrics used for measuring the classification performance of a method, indicating the effectiveness of my method for the problem of predicting hot spots.

An interesting extension of my method for predicting protein-protein interaction hot spots would be to predict hot spot interface residues for protein-nucleic acid complexes. Another application area of using my pipeline would be to predict multiple Post Translational Modification (PTM) types and sites [Yan+23]. In both these applications, the first step would be to calculate protein structure and sequence based features and then employ my pipeline to remove noise from the subsequent data matrices and then make the predictions.

Now, with the release of tools like AlphaFold-Multimer [Eva+21], the ease of generating *in silico* structures for protein complexes has significantly increased. Moreover, CM2D3 [Bot+23] is a workflow to generate protein-protein complex models. Indeed, AlphaFold-Multimer is one of the tools included in the CM2D3 workflow [Eva+21; Bot+23]. Thus, my method can be further extended by including CM2D3 workflow in the pipeline, from which sequence and structure-based features could be calculated in order to predict hotspots or PTM sites. This readily available structural and sequence based information of the protein complexes would further enable my method to make predictions for any protein bioinformatics problem that requires such information [CHW17].

My method can further be improved by using semi-supervised learning [CSZ09] (Chapter 3). One of the challenges of hot spot prediction using machine learning is the limited availability of experimental information on the energetic contribution of protein-protein interaction interface residues to the binding free energy of the complex. In other words, there is a limited number of ground truth interface residues that have been identified as hot spots or null spots. Ground truth labels for datasets are mostly annotated manually by a group of researchers and then later compared using different techniques to set target labels for the dataset. In hot spot prediction problem, the labeled residues are those that are identified with the help of experimental methods like ASM. Labeled data is important for a supervised machine learning method to make predictions. The existing approaches to use ML to predict hot spots are based on supervised learning [DPM07; Den+13; CKL09a; CKL09b; Ass+10].

In supervised learning, an algorithm is trained on input data that has been labeled for a specific output. The algorithm is trained to detect the underlying patterns and relationships between the input data and the output labels, enabling the algorithm to yield accurate labeling results when presented with never-before-seen data. Thus, insufficient labeled data makes it difficult to establish the relations between the input

and output.

In semi-supervised learning, the algorithm is trained on a dataset that contains both labeled and unlabeled data. Thus, a semi-supervised learning approach uses small amount of labeled data and large amounts of unlabeled data. This would decrease the cost and effort associated with experimentally annotating residues as hot spots or null spots using ASM. So, in SSL, the classifier is first trained on the labeled data, and then it is used to make predictions on the unlabeled data and also the probabilities for those predictions are generated. Now all the labels for predictions made with greater than a certain experimentally defined threshold of probability are now added to the existing training set. The classifier is then trained on this new training set and predictions are again made on the unlabeled data. This process is repeated until no more label predictions greater than threshold probability exist, or no unlabeled data remains.

In my thesis, as mentioned before, I have used classification algorithms, including Support vector machines, Random forests, Gradient Boosting machines and Extreme gradient boosting for training and predicting hot spots in a supervised fashion. Versions of these classification algorithms that employ semi supervised learning for making predictions are available [BD98; Lei+09; Mal+08] and can be added to my pipeline to make use of the unlabeled interface residues and further improve my method.

In conclusion, my method, namely, **R**obust principal component analysis **B**ased prediction of protein-protein interaction **H**ot **S**pots (RBHS), is a reliable method for predicting protein-protein interaction hot spot residues. Moreover, the work that I have presented in this thesis is an important step towards solving the issue of noise often present in biological data repositories, predicting not just protein-protein interaction hot spot residues, but also protein-nucleic acid hot spots, and finally defining independent training and testing sets that contain no overlap.

7 Supplementary Information

7.1 HB-34 Dataset

The interface residues in the dataset HB-34 are reported here. This table includes the PDB ID of the complex, the ID of the residue and label that denotes whether the residue is a hot spot or a null spot. Label=1, indicates the residue is a hot spot and label=0, indicates the residue is a null spot.

Table 7.1.1: Interface residues for HB-34 dataset.

S.No.	**PDB**	**Residue ID**	**Label**
1	1DFJI	202	0
2	1DFJI	257	0
3	1DFJI	259	1
4	1DFJI	283	0
5	1DFJI	285	0
6	1DFJI	314	0
7	1DFJI	316	0
8	1DFJI	340	0
9	1DFJI	397	0
10	1DFJI	430	1
11	1DFJI	431	1
12	1DFJI	433	1
13	1DFJI	453	0
14	1DFJI	455	0
15	2PCCA	197	1
16	2PCCA	290	1
17	1JTGA	110	1
18	1JCKB	23	1
19	1JCKB	60	0
20	1JCKB	90	1
21	1JCKB	91	1
22	1JCKB	103	0
23	1JCKB	176	1
24	1JCKB	210	1
25	1JCKA	90	1

Continued on next page

Table 7.1.1 – continued from previous page

S.No.	PDB	Residue ID	Label
26	1DANH	20	1
27	1EAWA	217	1
28	1C08B	32	1
29	1C08B	33	1
30	1C08B	53	1
31	1C08B	98	1
32	1C08A	31	1
33	1C08A	32	1
34	1C08A	96	1
35	1DANT	15	0
36	1DANT	17	0
37	1DANT	18	0
38	1DANT	20	1
39	1DANT	22	0
40	1DANT	24	0
41	1DANT	37	0
42	1DANT	41	0
43	1DANT	42	0
44	1DANT	43	0
45	1DANT	44	0
46	1DANT	45	0
47	1DANT	46	0
48	1DANT	47	0
49	1DANT	48	0
50	1DANT	51	0
51	1DANT	58	1
52	1DANT	60	1
53	1DANT	61	0
54	1DANT	62	0
55	1DANT	72	0
56	1DANT	76	0
57	1DANT	78	0
58	1A22B	243	1
59	1A22B	270	0
60	1A22B	274	0
61	1A22B	275	0
62	1A22B	280	0
63	1A22B	298	0
64	1A22B	301	0
65	1A22B	302	0
66	1A22B	303	0

Continued on next page

Table 7.1.1 – continued from previous page

S.No.	PDB	Residue ID	Label
67	1A22B	304	1
68	1A22B	305	0
69	1A22B	306	1
70	1A22B	320	0
71	1A22B	321	0
72	1A22B	324	0
73	1A22B	326	0
74	1A22B	327	0
75	1A22B	365	1
76	1A22B	366	0
77	1A22B	367	0
78	1A22B	369	1
79	1A22B	371	0
80	1A22B	417	0
81	1A22B	419	0
82	1IARB	13	1
83	1IARB	39	1
84	1IARB	41	1
85	1IARB	67	1
86	1IARB	69	1
87	1IARB	72	1
88	1IARB	74	1
89	1IARB	127	1
90	1IARB	183	1
91	1GC1C	23	0
92	1GC1C	25	0
93	1GC1C	27	0
94	1GC1C	29	0
95	1GC1C	32	0
96	1GC1C	33	0
97	1GC1C	35	0
98	1GC1C	40	0
99	1GC1C	42	0
100	1GC1C	44	0
101	1GC1C	45	0
102	1GC1C	52	0
103	1GC1C	59	0
104	1GC1C	60	0
105	1GC1C	63	0
106	1GC1C	64	0
107	1GC1C	85	0

Continued on next page

Table 7.1.1 – continued from previous page

S.No.	**PDB**	**Residue ID**	**Label**
108	1A22A	18	0
109	1A22A	21	0
110	1A22A	22	0
111	1A22A	25	0
112	1A22A	26	0
113	1A22A	42	0
114	1A22A	45	0
115	1A22A	46	0
116	1A22A	51	0
117	1A22A	56	0
118	1A22A	62	0
119	1A22A	63	0
120	1A22A	65	0
121	1A22A	164	0
122	1A22A	167	0
123	1A22A	171	0
124	1A22A	172	1
125	1A22A	175	1
126	1A22A	176	0
127	1A22A	178	1
128	1A22A	179	0
129	1A22A	183	0
130	1JRHH	32	0
131	1JRHH	52	1
132	1JRHH	53	1
133	1JRHH	54	1
134	1JRHH	55	1
135	1JRHH	56	0
136	1JRHH	58	0
137	1JRHH	60	0
138	1JRHH	100	0
139	1JRHH	104	0
140	1JRHH	107	0
141	1JRHL	27	0
142	1JRHL	30	0
143	1JRHL	50	1
144	1JRHL	91	0
145	1JRHL	92	1
146	1JRHL	93	0
147	1JRHL	94	0
148	1JTGB	49	1

Continued on next page

Table 7.1.1 – continued from previous page

S.No.	PDB	Residue ID	Label
149	1JTGB	74	1
150	1JTGB	142	1
151	1AK4C	485	1
152	1AK4C	486	1
153	1AK4C	487	1
154	1AK4C	489	1
155	1AK4C	490	1
156	1AK4C	493	1
157	2O3BB	24	1
158	2O3BB	74	1
159	2O3BB	76	1
160	1DANL	39	0
161	1DANL	42	0
162	1DANL	69	0
163	1DANL	73	0
164	1DANL	77	0
165	1DANL	88	0
166	1DANL	92	0
167	1DANL	93	0
168	1DANL	94	0
169	1EMVB	75	1
170	1EMVB	86	1
171	3HFMY	15	0
172	3HFMY	20	1
173	3HFMY	21	0
174	3HFMY	63	0
175	3HFMY	73	0
176	3HFMY	75	0
177	3HFMY	89	0
178	3HFMY	96	1
179	3HFMY	97	1
180	3HFMY	98	0
181	3HFMY	100	0
182	3HFMY	101	1
183	1IARA	5	0
184	1IARA	6	0
185	1IARA	8	0
186	1IARA	13	0
187	1IARA	78	0
188	1IARA	81	0
189	1IARA	84	0

Continued on next page

Table 7.1.1 – continued from previous page

S.No.	**PDB**	**Residue ID**	**Label**
190	1IARA	85	0
191	1IARA	88	1
192	1H9DB	104	1
193	2J0TD	2	1
194	2J0TD	68	1
195	1A4YB	5	1
196	1A4YB	8	0
197	1A4YB	12	0
198	1A4YB	13	0
199	1A4YB	31	0
200	1A4YB	32	0
201	1A4YB	68	0
202	1A4YB	84	0
203	1A4YB	89	0
204	1A4YB	108	0
205	1A4YB	114	0
206	1DVFB	32	0
207	1DVFB	52	1
208	1DVFB	54	1
209	1DVFB	56	0
210	1DVFB	58	0
211	1DVFB	98	1
212	1DVFB	100	1
213	1DVFB	101	1
214	1NMBH	99	1
215	1DVFD	30	0
216	1DVFD	33	0
217	1DVFD	52	0
218	1DVFD	97	1
219	1DVFD	98	1
220	1DVFD	102	1
221	1BRSA	27	1
222	1BRSA	54	0
223	1BRSA	58	1
224	1BRSA	59	1
225	1BRSA	60	0
226	1BRSA	73	1
227	1BRSA	87	1
228	1BRSA	102	1
229	1DVFA	49	0
230	1DVFA	50	0

Continued on next page

Table 7.1.1 – continued from previous page

S.No.	PDB	Residue ID	Label
231	1DVFA	92	0
232	1KTZB	27	1
233	1KTZB	30	1
234	1KTZB	32	0
235	1KTZB	49	0
236	1KTZB	50	1
237	1KTZB	52	0
238	1KTZB	55	0
239	1KTZB	77	0
240	1KTZB	118	0
241	1KTZB	119	0
242	1JRHI	47	1
243	1JRHI	49	1
244	1JRHI	52	1
245	1JRHI	53	1
246	1JRHI	54	0
247	1JRHI	55	0
248	1JRHI	82	1
249	1JRHI	84	0
250	1JRHI	98	0
251	1BRSD	29	1
252	1BRSD	35	1
253	1BRSD	39	1
254	1BRSD	42	0
255	2JELP	70	1
256	1BXIA	27	0
257	1BXIA	28	0
258	1BXIA	29	0
259	1BXIA	33	1
260	1BXIA	34	1
261	1BXIA	37	0
262	1BXIA	38	0
263	1BXIA	41	1
264	1BXIA	46	0
265	1BXIA	48	0
266	1BXIA	50	1
267	1BXIA	51	1
268	1BXIA	53	0
269	1BXIA	54	1
270	1BXIA	55	1
271	1KTZA	94	1

Continued on next page

Table 7.1.1 – continued from previous page

S.No.	PDB	Residue ID	Label
272	2WPTA	37	1
273	2WPTA	41	1
274	2WPTA	50	1
275	2WPTA	56	1
276	1XD3B	8	1
277	1FFWB	214	1
278	1TM1I	58	1
279	1TM1I	60	1
280	1TM1I	61	1
281	1TM1I	65	1
282	1TM1I	67	1
283	1CBWD	11	0
284	1CBWD	15	1
285	1CBWD	17	0
286	1CBWD	19	0
287	1CBWD	34	0
288	1CBWD	39	0
289	1FCCC	25	0
290	1FCCC	27	1
291	1FCCC	28	0
292	1FCCC	31	1
293	1FCCC	35	1
294	1FCCC	40	0
295	1FCCC	43	1
296	1CHOI	17	1
297	1CHOI	18	1
298	1CHOI	19	1
299	1CHOI	20	1
300	1CHOI	21	1
301	1FC2C	147	0
302	1FC2C	150	1
303	1FC2C	154	0
304	1F47A	5	0
305	1F47A	6	0
306	1F47A	7	0
307	1F47A	8	1
308	1F47A	11	1
309	1F47A	12	1
310	1F47A	14	0
311	1F47A	15	0
312	1DN2E	10	1

Continued on next page

Table 7.1.1 – continued from previous page

S.No.	PDB	Residue ID	Label
313	1DN2E	11	1

7.2 BID-18 Dataset

The interface residues in the dataset BID-18 are reported here. This table includes the PDB ID of the complex, the ID of the residue and label that denotes whether the residue is a hot spot or a null spot. Label=1, indicates the residue is a hot spot and label=0, indicates the residue is a null spot.

Table 7.2.1: Interface residues for BID-18 dataset.

S.No.	PDB	Residue ID	Label
1	1CDLA	12	0
2	1CDLA	19	0
3	1CDLA	92	1
4	1CDLE	799	0
5	1CDLE	800	1
6	1CDLE	802	0
7	1CDLE	804	1
8	1CDLE	808	0
9	1CDLE	810	1
10	1CDLE	811	0
11	1CDLE	812	1
12	1CDLE	813	1
13	1DVAH	38	0
14	1DVAH	65	0
15	1DVAH	67	0
16	1DVAH	70	0
17	1DVAH	73	0
18	1DVAH	74	0
19	1DVAH	75	0
20	1DVAH	76	1
21	1DVAH	80	0
22	1DVAH	82	0
23	1DVAH	144	0
24	1DVAH	153	0
25	1DVAX	1	0

Continued on next page

Table 7.2.1 – continued from previous page

S.No.	PDB	Residue ID	Label
26	1DVAX	2	1
27	1DVAX	5	0
28	1DVAX	7	0
29	1DVAX	8	0
30	1DVAX	9	0
31	1DVAX	11	1
32	1DVAX	12	1
33	1DVAX	14	0
34	1DVAX	15	1
35	1DVAX	16	0
36	1DX5N	24	0
37	1DX5N	34	0
38	1DX5N	36	0
39	1DX5N	37	0
40	1DX5N	38	0
41	1DX5N	39	0
42	1DX5N	65	0
43	1DX5N	67	1
44	1DX5N	74	0
45	1DX5N	75	0
46	1DX5N	76	1
47	1DX5N	80	1
48	1DX5N	81	0
49	1DX5N	82	0
50	1DX5N	84	0
51	1DX5N	110	0
52	1DX5N	235	0
53	1EBPA	93	1
54	1EBPA	150	1
55	1EBPA	151	0
56	1EBPA	205	1
57	1EBPC	9	0
58	1EBPC	10	0
59	1EBPC	11	0
60	1EBPC	12	0
61	1EBPC	13	1
62	1ES7A	26	0
63	1ES7A	31	1
64	1ES7A	49	0
65	1ES7A	50	0
66	1FAKT	15	0

Continued on next page

Table 7.2.1 – continued from previous page

S.No.	PDB	Residue ID	Label
67	1FAKT	17	0
68	1FAKT	18	0
69	1FAKT	20	1
70	1FAKT	22	0
71	1FAKT	24	0
72	1FAKT	37	0
73	1FAKT	41	0
74	1FAKT	42	0
75	1FAKT	44	0
76	1FAKT	47	0
77	1FAKT	48	0
78	1FAKT	50	0
79	1FAKT	58	1
80	1FAKT	94	0
81	1FAKT	128	0
82	1FAKT	133	0
83	1FAKT	135	0
84	1FAKT	140	0
85	1FAKT	203	0
86	1FAKT	207	0
87	1FE8A	963	0
88	1FE8A	987	0
89	1FE8A	990	0
90	1FE8A	1023	0
91	1FOEB	41	0
92	1FOEB	54	1
93	1G3IA	438	1
94	1G3IA	439	1
95	1G3IA	441	1
96	1G3IA	442	1
97	1G3IA	443	1
98	1G3IA	444	1
99	1GL4A	403	0
100	1GL4A	427	1
101	1GL4A	429	1
102	1GL4A	431	1
103	1GL4A	440	0
104	1GL4A	616	1
105	1GL4A	620	1
106	1IHBB	101	0
107	1IHBB	133	0

Continued on next page

Table 7.2.1 – continued from previous page

S.No.	PDB	Residue ID	Label
108	1IHBB	135	0
109	1IHBB	136	0
110	1JATA	55	1
111	1JATB	8	1
112	1JPPB	345	1
113	1JPPB	354	0
114	1JPPB	383	1
115	1JPPB	386	0
116	1JPPB	435	0
117	1JPPB	469	0
118	1JPPB	470	0
119	1MQ8B	206	1
120	1NFIF	181	1
121	1NFIF	215	0
122	1NUNA	76	0
123	1NUNA	78	0
124	1NUNA	155	0
125	1UB4C	453	0
126	2HHBB	35	0

7.3 Blosum62 features for HB-34

This table includes the Block substitution matrix (Blosum62) features calculated for the residues in HB-34 dataset. These letters here indicate the Amino Acid Type.

Table 7.3.1: Blosum62 features for HB-34 dataset.

S.No.	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
1	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
2	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
3	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
4	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
5	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
6	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
7	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
8	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
9	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
10	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
11	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
12	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
13	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
14	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
15	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1
16	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
17	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
18	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
19	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
20	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
21	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1
22	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
23	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
24	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
25	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
26	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2

Continued on next page

Table 7.3.1 – continued from previous page

S.No.	**A**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**K**	**L**	**M**	**N**	**P**	**Q**	**R**	**S**	**T**	**V**	**W**	**Y**
27	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
28	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
29	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
30	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
31	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
32	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
33	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
34	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
35	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
36	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
37	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
38	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
39	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
40	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
41	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
42	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
43	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
44	0	-3	-1	-2	-3	6	-2	-4	-2	-4	-3	0	-2	-2	-2	0	-2	-3	-2	-3
45	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
46	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
47	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
48	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
49	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
50	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
51	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
52	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
53	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3

Continued on next page

Table 7.3.1 – continued from previous page

S.No.	**A**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**K**	**L**	**M**	**N**	**P**	**Q**	**R**	**S**	**T**	**V**	**W**	**Y**
54	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
55	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
56	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
57	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
58	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
59	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
60	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
61	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
62	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
63	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
64	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
65	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
66	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
67	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
68	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
69	-1	-3	-1	-1	-4	-2	-2	-3	-1	-3	-2	-2	7	-1	-2	-1	-1	-2	-4	-3
70	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
71	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
72	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
73	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
74	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
75	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
76	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
77	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
78	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
79	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1
80	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2

Continued on next page

Table 7.3.1 – continued from previous page

S.No.	**A**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**K**	**L**	**M**	**N**	**P**	**Q**	**R**	**S**	**T**	**V**	**W**	**Y**
81	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
82	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
83	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
84	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
85	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
86	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1
87	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
88	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
89	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
90	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
91	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
92	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
93	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
94	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
95	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
96	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
97	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
98	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
99	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
100	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
101	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
102	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
103	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
104	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
105	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
106	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
107	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2

Continued on next page

Table 7.3.1 – continued from previous page

S.No.	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
108	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
109	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
110	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
111	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
112	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
113	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
114	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
115	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
116	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
117	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
118	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
119	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
120	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
121	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
122	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
123	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
124	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
125	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
126	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
127	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
128	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
129	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
130	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
131	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
132	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
133	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
134	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3

Continued on next page

Table 7.3.1 – continued from previous page

S.No.	**A**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**K**	**L**	**M**	**N**	**P**	**Q**	**R**	**S**	**T**	**V**	**W**	**Y**
135	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
136	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
137	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
138	0	-3	-1	-2	-3	6	-2	-4	-2	-4	-3	0	-2	-2	-2	0	-2	-3	-2	-3
139	0	-3	-1	-2	-3	6	-2	-4	-2	-4	-3	0	-2	-2	-2	0	-2	-3	-2	-3
140	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
141	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
142	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
143	0	-3	-1	-2	-3	6	-2	-4	-2	-4	-3	0	-2	-2	-2	0	-2	-3	-2	-3
144	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
145	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
146	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
147	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
148	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
149	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
150	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
151	-1	-3	-1	-1	-4	-2	-2	-3	-1	-3	-2	-2	7	-1	-2	-1	-1	-2	-4	-3
152	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1
153	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
154	0	-3	-1	-2	-3	6	-2	-4	-2	-4	-3	0	-2	-2	-2	0	-2	-3	-2	-3
155	-1	-3	-1	-1	-4	-2	-2	-3	-1	-3	-2	-2	7	-1	-2	-1	-1	-2	-4	-3
156	-1	-3	-1	-1	-4	-2	-2	-3	-1	-3	-2	-2	7	-1	-2	-1	-1	-2	-4	-3
157	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
158	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
159	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
160	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
161	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1

Continued on next page

Table 7.3.1 – continued from previous page

S.No.	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
162	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
163	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
164	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
165	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
166	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1
167	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
168	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
169	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
170	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
171	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
172	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
173	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
174	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
175	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
176	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
177	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
178	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
179	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
180	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
181	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
182	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
183	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
184	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
185	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
186	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
187	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
188	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2

Continued on next page

Table 7.3.1 – continued from previous page

S.No.	**A**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**K**	**L**	**M**	**N**	**P**	**Q**	**R**	**S**	**T**	**V**	**W**	**Y**
189	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
190	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
191	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
192	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
193	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
194	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
195	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
196	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
197	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
198	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
199	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
200	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
201	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
202	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
203	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
204	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
205	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
206	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
207	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
208	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
209	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
210	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
211	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
212	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
213	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
214	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
215	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2

Continued on next page

Table 7.3.1 – continued from previous page

S.No.	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
216	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
217	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
218	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
219	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
220	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
221	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
222	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
223	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
224	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
225	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
226	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
227	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
228	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
229	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
230	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
231	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
232	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
233	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
234	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
235	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
236	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
237	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
238	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
239	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1
240	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
241	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
242	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2

Continued on next page

Table 7.3.1 – continued from previous page

S.No.	**A**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**K**	**L**	**M**	**N**	**P**	**Q**	**R**	**S**	**T**	**V**	**W**	**Y**
243	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
244	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
245	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
246	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
247	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
248	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
249	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
250	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
251	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
252	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
253	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
254	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
255	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
256	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
257	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
258	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
259	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
260	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1
261	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1
262	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
263	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
264	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
265	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
266	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
267	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
268	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
269	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7

Continued on next page

Table 7.3.1 – continued from previous page

S.No.	**A**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**K**	**L**	**M**	**N**	**P**	**Q**	**R**	**S**	**T**	**V**	**W**	**Y**
270	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
271	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
272	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1
273	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
274	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
275	-1	-3	-1	-1	-4	-2	-2	-3	-1	-3	-2	-2	7	-1	-2	-1	-1	-2	-4	-3
276	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
277	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
278	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
279	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
280	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
281	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
282	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
283	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
284	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
285	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
286	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
287	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1
288	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
289	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
290	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
291	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
292	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
293	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
294	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
295	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
296	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2

Continued on next page

Table 7.3.1 – continued from previous page

S.No.	**A**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**K**	**L**	**M**	**N**	**P**	**Q**	**R**	**S**	**T**	**V**	**W**	**Y**
297	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
298	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
299	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
300	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
301	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
302	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
303	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
304	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
305	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
306	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
307	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
308	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
309	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
310	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
311	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
312	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1
313	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2

7.4 Blosum62 features for BID-18

This table includes the Block substitution matrix(Blosum62) features calculated for the residues in BID-18 dataset. These letters here indicate the Amino Acid Type.

Table 7.4.1: Blosum62 features for BID-18 dataset.

S.No.	**A**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**K**	**L**	**M**	**N**	**P**	**Q**	**R**	**S**	**T**	**V**	**W**	**Y**
1	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
2	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
3	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
4	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
5	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
6	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
7	0	-3	-1	-2	-3	6	-2	-4	-2	-4	-3	0	-2	-2	-2	0	-2	-3	-2	-3
8	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
9	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
10	0	-3	-1	-2	-3	6	-2	-4	-2	-4	-3	0	-2	-2	-2	0	-2	-3	-2	-3
11	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
12	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
13	0	-3	-1	-2	-3	6	-2	-4	-2	-4	-3	0	-2	-2	-2	0	-2	-3	-2	-3
14	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
15	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1
16	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
17	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
18	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
19	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
20	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
21	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
22	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
23	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
24	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
25	4	0	-2	-1	-2	0	-2	-1	-1	-1	-1	-2	-1	-1	-1	1	0	0	-3	-2
26	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1

Continued on next page

Table 7.4.1 – continued from previous page

S.No.	**A**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**K**	**L**	**M**	**N**	**P**	**Q**	**R**	**S**	**T**	**V**	**W**	**Y**
27	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
28	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
29	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1
30	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
31	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
32	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
33	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
34	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
35	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1
36	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
37	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
38	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
39	-1	-3	-1	-1	-4	-2	-2	-3	-1	-3	-2	-2	7	-1	-2	-1	-1	-2	-4	-3
40	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
41	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
42	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
43	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
44	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
45	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
46	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
47	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
48	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
49	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
50	-1	-1	-3	-2	0	-3	-2	1	-1	2	5	-2	-2	0	-1	-1	-1	1	-1	-1
51	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
52	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
53	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3

Continued on next page

Table 7.4.1 – continued from previous page

S.No.	**A**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**K**	**L**	**M**	**N**	**P**	**Q**	**R**	**S**	**T**	**V**	**W**	**Y**
54	-1	-1	-3	-2	0	-3	-2	1	-1	2	5	-2	-2	0	-1	-1	-1	1	-1	-1
55	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
56	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
57	0	-3	-1	-2	-3	6	-2	-4	-2	-4	-3	0	-2	-2	-2	0	-2	-3	-2	-3
58	-1	-3	-1	-1	-4	-2	-2	-3	-1	-3	-2	-2	7	-1	-2	-1	-1	-2	-4	-3
59	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
60	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
61	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
62	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1
63	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
64	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
65	-1	-3	-1	-1	-4	-2	-2	-3	-1	-3	-2	-2	7	-1	-2	-1	-1	-2	-4	-3
66	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
67	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
68	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
69	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
70	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
71	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
72	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
73	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
74	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
75	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
76	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
77	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
78	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
79	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
80	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7

Continued on next page

Table 7.4.1 – continued from previous page

S.No.	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
81	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
82	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
83	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
84	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
85	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
86	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1
87	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
88	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
89	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
90	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
91	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
92	0	-3	-1	-2	-3	6	-2	-4	-2	-4	-3	0	-2	-2	-2	0	-2	-3	-2	-3
93	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
94	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
95	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
96	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
97	-1	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
98	-1	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
99	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
100	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
101	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
102	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
103	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
104	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
105	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
106	-2	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-2
107	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2

Continued on next page

Table 7.4.1 – continued from previous page

S.No.	**A**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**K**	**L**	**M**	**N**	**P**	**Q**	**R**	**S**	**T**	**V**	**W**	**Y**
108	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
109	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
110	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
111	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
112	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
113	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
114	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
115	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
116	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
117	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
118	-2	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
119	0	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-2
120	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7
121	0	9	-3	-4	-2	-3	-3	-1	-3	-1	-1	-3	-3	-3	-3	-1	-1	-1	-2	-2
122	-2	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
123	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
124	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-2
125	-2	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
126	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	7

7.5 Physicochemical features for HB-34

This table includes the six physicochemical features calculated from the AAIndex database for the residues in HB-34 dataset except for propensities which were calculated from [JT97; JT96]. AASA is Average Accessible Surface area.

Table 7.5.1: Physicochemical features for HB-34 dataset.

S.No.	Hydrophobicity	Hydrophilicity	Polarity	Polarizability	Propensities	AASA
1	-0.62	3	12.3	0.15	-0.13	68.2
2	0.37	-3.4	5.4	0.41	0.83	34.7
3	0.37	-3.4	5.4	0.41	0.83	34.7
4	-0.62	3	12.3	0.15	-0.13	68.2
5	-0.26	0.3	9.2	0.06	-0.33	42
6	0.37	-3.4	5.4	0.41	0.83	34.7
7	-1.1	3	11.3	0.22	-0.36	103
8	-0.62	3	12.3	0.15	-0.13	68.2
9	-0.62	3	12.3	0.15	-0.13	68.2
10	0.02	-2.3	6.2	0.3	0.66	55.2
11	-0.72	3	13	0.11	-0.38	60.6
12	0.02	-2.3	6.2	0.3	0.66	55.2
13	-1.76	3	10.5	0.29	0.27	94.7
14	0.73	-1.8	5.2	0.19	0.44	22.8
15	0.54	-1.5	5.9	0.14	0.27	23.7
16	-0.62	3	12.3	0.15	-0.13	68.2
17	-0.62	3	12.3	0.15	-0.13	68.2
18	-0.64	2	11.6	0.13	0.12	60.1
19	-0.64	2	11.6	0.13	0.12	60.1
20	0.02	-2.3	6.2	0.3	0.66	55.2
21	0.54	-1.5	5.9	0.14	0.27	23.7
22	-1.1	3	11.3	0.22	-0.36	103
23	0.61	-2.5	5.2	0.29	0.82	25.5
24	-0.69	0.2	10.5	0.18	-0.11	68.7
25	0.02	-2.3	6.2	0.3	0.66	55.2
26	-1.1	3	11.3	0.22	-0.36	103
27	-0.72	3	13	0.11	-0.38	60.6
28	-0.72	3	13	0.11	-0.38	60.6
29	0.02	-2.3	6.2	0.3	0.66	55.2
30	0.02	-2.3	6.2	0.3	0.66	55.2
31	0.37	-3.4	5.4	0.41	0.83	34.7
32	-0.64	2	11.6	0.13	0.12	60.1
33	-0.64	2	11.6	0.13	0.12	60.1
34	0.02	-2.3	6.2	0.3	0.66	55.2

Continued on next page

Table 7.5.1 – continued from previous page

S.No.	Hydrophobicity	Hydrophilicity	Polarity	Polarizability	Propensities	AASA
35	-1.1	3	11.3	0.22	-0.36	103
36	-0.18	-0.4	8.6	0.11	-0.18	45
37	-0.64	2	11.6	0.13	0.12	60.1
38	-1.1	3	11.3	0.22	-0.36	103
39	0.73	-1.8	5.2	0.19	0.44	22.8
40	-0.62	3	12.3	0.15	-0.13	68.2
41	-0.69	0.2	10.5	0.18	-0.11	68.7
42	-1.1	3	11.3	0.22	-0.36	103
43	-0.26	0.3	9.2	0.06	-0.33	42
44	0.16	0	9	0	-0.07	24.5
45	-0.72	3	13	0.11	-0.38	60.6
46	0.37	-3.4	5.4	0.41	0.83	34.7
47	-1.1	3	11.3	0.22	-0.36	103
48	-0.26	0.3	9.2	0.06	-0.33	42
49	-1.1	3	11.3	0.22	-0.36	103
50	0.02	-2.3	6.2	0.3	0.66	55.2
51	-0.72	3	13	0.11	-0.38	60.6
52	-0.18	-0.4	8.6	0.11	-0.18	45
53	-0.72	3	13	0.11	-0.38	60.6
54	-0.62	3	12.3	0.15	-0.13	68.2
55	0.53	-1.8	4.9	0.19	0.4	27.6
56	0.61	-2.5	5.2	0.29	0.82	25.5
57	0.02	-2.3	6.2	0.3	0.66	55.2
58	-1.76	3	10.5	0.29	0.27	94.7
59	-1.76	3	10.5	0.29	0.27	94.7
60	-0.69	0.2	10.5	0.18	-0.11	68.7
61	-0.62	3	12.3	0.15	-0.13	68.2
62	0.37	-3.4	5.4	0.41	0.83	34.7
63	-0.26	0.3	9.2	0.06	-0.33	42
64	-0.18	-0.4	8.6	0.11	-0.18	45
65	-0.26	0.3	9.2	0.06	-0.33	42
66	0.73	-1.8	5.2	0.19	0.44	22.8
67	0.37	-3.4	5.4	0.41	0.83	34.7
68	0.73	-1.8	5.2	0.19	0.44	22.8
69	-0.07	0	8	0.13	-0.25	51.5
70	-0.62	3	12.3	0.15	-0.13	68.2
71	-1.1	3	11.3	0.22	-0.36	103
72	-0.26	0.3	9.2	0.06	-0.33	42
73	-0.72	3	13	0.11	-0.38	60.6
74	-0.62	3	12.3	0.15	-0.13	68.2
75	0.73	-1.8	5.2	0.19	0.44	22.8

Continued on next page

Table 7.5.1 – continued from previous page

S.No.	Hydrophobicity	Hydrophilicity	Polarity	Polarizability	Propensities	AASA
76	-0.69	0.2	10.5	0.18	-0.11	68.7
77	-1.1	3	11.3	0.22	-0.36	103
78	0.37	-3.4	5.4	0.41	0.83	34.7
79	0.54	-1.5	5.9	0.14	0.27	23.7
80	-1.76	3	10.5	0.29	0.27	94.7
81	-0.26	0.3	9.2	0.06	-0.33	42
82	0.02	-2.3	6.2	0.3	0.66	55.2
83	0.53	-1.8	4.9	0.19	0.4	27.6
84	0.61	-2.5	5.2	0.29	0.82	25.5
85	-0.72	3	13	0.11	-0.38	60.6
86	0.54	-1.5	5.9	0.14	0.27	23.7
87	-0.72	3	13	0.11	-0.38	60.6
88	0.02	-2.3	6.2	0.3	0.66	55.2
89	0.02	-2.3	6.2	0.3	0.66	55.2
90	0.02	-2.3	6.2	0.3	0.66	55.2
91	-0.26	0.3	9.2	0.06	-0.33	42
92	-0.69	0.2	10.5	0.18	-0.11	68.7
93	-0.4	-0.5	10.4	0.23	0.41	50.7
94	-1.1	3	11.3	0.22	-0.36	103
95	-0.64	2	11.6	0.13	0.12	60.1
96	-0.69	0.2	10.5	0.18	-0.11	68.7
97	-1.1	3	11.3	0.22	-0.36	103
98	-0.69	0.2	10.5	0.18	-0.11	68.7
99	-0.26	0.3	9.2	0.06	-0.33	42
100	0.53	-1.8	4.9	0.19	0.4	27.6
101	-0.18	-0.4	8.6	0.11	-0.18	45
102	-0.64	2	11.6	0.13	0.12	60.1
103	-1.76	3	10.5	0.29	0.27	94.7
104	-0.26	0.3	9.2	0.06	-0.33	42
105	-0.72	3	13	0.11	-0.38	60.6
106	-0.69	0.2	10.5	0.18	-0.11	68.7
107	-0.62	3	12.3	0.15	-0.13	68.2
108	-0.4	-0.5	10.4	0.23	0.41	50.7
109	-0.4	-0.5	10.4	0.23	0.41	50.7
110	-0.69	0.2	10.5	0.18	-0.11	68.7
111	0.61	-2.5	5.2	0.29	0.82	25.5
112	-0.72	3	13	0.11	-0.38	60.6
113	0.02	-2.3	6.2	0.3	0.66	55.2
114	0.53	-1.8	4.9	0.19	0.4	27.6
115	-0.69	0.2	10.5	0.18	-0.11	68.7
116	-0.26	0.3	9.2	0.06	-0.33	42

Continued on next page

Table 7.5.1 – continued from previous page

S.No.	Hydrophobicity	Hydrophilicity	Polarity	Polarizability	Propensities	AASA
117	-0.62	3	12.3	0.15	-0.13	68.2
118	-0.26	0.3	9.2	0.06	-0.33	42
119	-0.64	2	11.6	0.13	0.12	60.1
120	-0.62	3	12.3	0.15	-0.13	68.2
121	0.02	-2.3	6.2	0.3	0.66	55.2
122	-1.76	3	10.5	0.29	0.27	94.7
123	-0.72	3	13	0.11	-0.38	60.6
124	-1.1	3	11.3	0.22	-0.36	103
125	-0.18	-0.4	8.6	0.11	-0.18	45
126	0.61	-2.5	5.2	0.29	0.82	25.5
127	-1.76	3	10.5	0.29	0.27	94.7
128	0.73	-1.8	5.2	0.19	0.44	22.8
129	-1.76	3	10.5	0.29	0.27	94.7
130	0.02	-2.3	6.2	0.3	0.66	55.2
131	0.37	-3.4	5.4	0.41	0.83	34.7
132	0.37	-3.4	5.4	0.41	0.83	34.7
133	-0.72	3	13	0.11	-0.38	60.6
134	-0.72	3	13	0.11	-0.38	60.6
135	-0.72	3	13	0.11	-0.38	60.6
136	0.02	-2.3	6.2	0.3	0.66	55.2
137	-0.64	2	11.6	0.13	0.12	60.1
138	0.16	0	9	0	-0.07	24.5
139	0.16	0	9	0	-0.07	24.5
140	-0.18	-0.4	8.6	0.11	-0.18	45
141	-0.62	3	12.3	0.15	-0.13	68.2
142	0.02	-2.3	6.2	0.3	0.66	55.2
143	0.16	0	9	0	-0.07	24.5
144	0.02	-2.3	6.2	0.3	0.66	55.2
145	0.37	-3.4	5.4	0.41	0.83	34.7
146	-0.26	0.3	9.2	0.06	-0.33	42
147	-0.18	-0.4	8.6	0.11	-0.18	45
148	-0.72	3	13	0.11	-0.38	60.6
149	-1.1	3	11.3	0.22	-0.36	103
150	0.61	-2.5	5.2	0.29	0.82	25.5
151	-0.07	0	8	0.13	-0.25	51.5
152	0.54	-1.5	5.9	0.14	0.27	23.7
153	-0.4	-0.5	10.4	0.23	0.41	50.7
154	0.16	0	9	0	-0.07	24.5
155	-0.07	0	8	0.13	-0.25	51.5
156	-0.07	0	8	0.13	-0.25	51.5
157	-0.62	3	12.3	0.15	-0.13	68.2

Continued on next page

Table 7.5.1 – continued from previous page

S.No.	Hydrophobicity	Hydrophilicity	Polarity	Polarizability	Propensities	AASA
158	-0.69	0.2	10.5	0.18	-0.11	68.7
159	0.37	-3.4	5.4	0.41	0.83	34.7
160	0.53	-1.8	4.9	0.19	0.4	27.6
161	0.73	-1.8	5.2	0.19	0.44	22.8
162	0.73	-1.8	5.2	0.19	0.44	22.8
163	0.53	-1.8	4.9	0.19	0.4	27.6
164	-0.62	3	12.3	0.15	-0.13	68.2
165	-0.69	0.2	10.5	0.18	-0.11	68.7
166	0.54	-1.5	5.9	0.14	0.27	23.7
167	-0.64	2	11.6	0.13	0.12	60.1
168	-0.62	3	12.3	0.15	-0.13	68.2
169	-0.64	2	11.6	0.13	0.12	60.1
170	0.61	-2.5	5.2	0.29	0.82	25.5
171	-0.4	-0.5	10.4	0.23	0.41	50.7
172	0.02	-2.3	6.2	0.3	0.66	55.2
173	-1.76	3	10.5	0.29	0.27	94.7
174	0.37	-3.4	5.4	0.41	0.83	34.7
175	-1.76	3	10.5	0.29	0.27	94.7
176	0.53	-1.8	4.9	0.19	0.4	27.6
177	-0.18	-0.4	8.6	0.11	-0.18	45
178	-1.1	3	11.3	0.22	-0.36	103
179	-1.1	3	11.3	0.22	-0.36	103
180	0.73	-1.8	5.2	0.19	0.44	22.8
181	-0.26	0.3	9.2	0.06	-0.33	42
182	-0.72	3	13	0.11	-0.38	60.6
183	0.73	-1.8	5.2	0.19	0.44	22.8
184	-0.18	-0.4	8.6	0.11	-0.18	45
185	-0.69	0.2	10.5	0.18	-0.11	68.7
186	-0.18	-0.4	8.6	0.11	-0.18	45
187	-0.69	0.2	10.5	0.18	-0.11	68.7
188	-1.76	3	10.5	0.29	0.27	94.7
189	-1.1	3	11.3	0.22	-0.36	103
190	-1.76	3	10.5	0.29	0.27	94.7
191	-1.76	3	10.5	0.29	0.27	94.7
192	-0.64	2	11.6	0.13	0.12	60.1
193	-0.18	-0.4	8.6	0.11	-0.18	45
194	-0.26	0.3	9.2	0.06	-0.33	42
195	-1.76	3	10.5	0.29	0.27	94.7
196	-0.4	-0.5	10.4	0.23	0.41	50.7
197	-0.69	0.2	10.5	0.18	-0.11	68.7
198	-0.4	-0.5	10.4	0.23	0.41	50.7

Continued on next page

Table 7.5.1 – continued from previous page

S.No.	Hydrophobicity	Hydrophilicity	Polarity	Polarizability	Propensities	AASA
199	-1.76	3	10.5	0.29	0.27	94.7
200	-1.76	3	10.5	0.29	0.27	94.7
201	-0.64	2	11.6	0.13	0.12	60.1
202	-0.4	-0.5	10.4	0.23	0.41	50.7
203	0.37	-3.4	5.4	0.41	0.83	34.7
204	-0.62	3	12.3	0.15	-0.13	68.2
205	-0.4	-0.5	10.4	0.23	0.41	50.7
206	0.02	-2.3	6.2	0.3	0.66	55.2
207	0.37	-3.4	5.4	0.41	0.83	34.7
208	-0.72	3	13	0.11	-0.38	60.6
209	-0.64	2	11.6	0.13	0.12	60.1
210	-0.72	3	13	0.11	-0.38	60.6
211	-0.62	3	12.3	0.15	-0.13	68.2
212	-0.72	3	13	0.11	-0.38	60.6
213	0.02	-2.3	6.2	0.3	0.66	55.2
214	0.02	-2.3	6.2	0.3	0.66	55.2
215	-1.1	3	11.3	0.22	-0.36	103
216	-0.4	-0.5	10.4	0.23	0.41	50.7
217	-0.72	3	13	0.11	-0.38	60.6
218	0.73	-1.8	5.2	0.19	0.44	22.8
219	0.02	-2.3	6.2	0.3	0.66	55.2
220	0.02	-2.3	6.2	0.3	0.66	55.2
221	-1.1	3	11.3	0.22	-0.36	103
222	-0.72	3	13	0.11	-0.38	60.6
223	-0.64	2	11.6	0.13	0.12	60.1
224	-1.76	3	10.5	0.29	0.27	94.7
225	-0.62	3	12.3	0.15	-0.13	68.2
226	-0.62	3	12.3	0.15	-0.13	68.2
227	-1.76	3	10.5	0.29	0.27	94.7
228	-0.4	-0.5	10.4	0.23	0.41	50.7
229	0.02	-2.3	6.2	0.3	0.66	55.2
230	0.02	-2.3	6.2	0.3	0.66	55.2
231	0.37	-3.4	5.4	0.41	0.83	34.7
232	0.53	-1.8	4.9	0.19	0.4	27.6
233	0.61	-2.5	5.2	0.29	0.82	25.5
234	-0.72	3	13	0.11	-0.38	60.6
235	-0.26	0.3	9.2	0.06	-0.33	42
236	0.73	-1.8	5.2	0.19	0.44	22.8
237	-0.26	0.3	9.2	0.06	-0.33	42
238	-0.62	3	12.3	0.15	-0.13	68.2
239	0.54	-1.5	5.9	0.14	0.27	23.7

Continued on next page

Table 7.5.1 – continued from previous page

S.No.	Hydrophobicity	Hydrophilicity	Polarity	Polarizability	Propensities	AASA
240	-0.72	3	13	0.11	-0.38	60.6
241	-0.62	3	12.3	0.15	-0.13	68.2
242	-1.1	3	11.3	0.22	-0.36	103
243	0.02	-2.3	6.2	0.3	0.66	55.2
244	-1.1	3	11.3	0.22	-0.36	103
245	-0.64	2	11.6	0.13	0.12	60.1
246	-0.26	0.3	9.2	0.06	-0.33	42
247	-0.62	3	12.3	0.15	-0.13	68.2
248	0.37	-3.4	5.4	0.41	0.83	34.7
249	-1.76	3	10.5	0.29	0.27	94.7
250	-1.1	3	11.3	0.22	-0.36	103
251	0.02	-2.3	6.2	0.3	0.66	55.2
252	-0.72	3	13	0.11	-0.38	60.6
253	-0.72	3	13	0.11	-0.38	60.6
254	-0.18	-0.4	8.6	0.11	-0.18	45
255	-0.62	3	12.3	0.15	-0.13	68.2
256	-0.18	-0.4	8.6	0.11	-0.18	45
257	-0.26	0.3	9.2	0.06	-0.33	42
258	-0.26	0.3	9.2	0.06	-0.33	42
259	0.53	-1.8	4.9	0.19	0.4	27.6
260	0.54	-1.5	5.9	0.14	0.27	23.7
261	0.54	-1.5	5.9	0.14	0.27	23.7
262	-0.18	-0.4	8.6	0.11	-0.18	45
263	-0.62	3	12.3	0.15	-0.13	68.2
264	-0.4	-0.5	10.4	0.23	0.41	50.7
265	-0.26	0.3	9.2	0.06	-0.33	42
266	-0.26	0.3	9.2	0.06	-0.33	42
267	-0.72	3	13	0.11	-0.38	60.6
268	0.73	-1.8	5.2	0.19	0.44	22.8
269	0.02	-2.3	6.2	0.3	0.66	55.2
270	0.02	-2.3	6.2	0.3	0.66	55.2
271	-1.76	3	10.5	0.29	0.27	94.7
272	0.54	-1.5	5.9	0.14	0.27	23.7
273	-0.62	3	12.3	0.15	-0.13	68.2
274	-0.26	0.3	9.2	0.06	-0.33	42
275	-0.07	0	8	0.13	-0.25	51.5
276	0.53	-1.8	4.9	0.19	0.4	27.6
277	0.61	-2.5	5.2	0.29	0.82	25.5
278	-0.18	-0.4	8.6	0.11	-0.18	45
279	-0.62	3	12.3	0.15	-0.13	68.2
280	0.02	-2.3	6.2	0.3	0.66	55.2

Continued on next page

Table 7.5.1 – continued from previous page

S.No.	Hydrophobicity	Hydrophilicity	Polarity	Polarizability	Propensities	AASA
281	-1.76	3	10.5	0.29	0.27	94.7
282	-1.76	3	10.5	0.29	0.27	94.7
283	-0.18	-0.4	8.6	0.11	-0.18	45
284	-1.1	3	11.3	0.22	-0.36	103
285	-1.76	3	10.5	0.29	0.27	94.7
286	0.73	-1.8	5.2	0.19	0.44	22.8
287	0.54	-1.5	5.9	0.14	0.27	23.7
288	-1.76	3	10.5	0.29	0.27	94.7
289	-0.18	-0.4	8.6	0.11	-0.18	45
290	-0.62	3	12.3	0.15	-0.13	68.2
291	-1.1	3	11.3	0.22	-0.36	103
292	-1.1	3	11.3	0.22	-0.36	103
293	-0.64	2	11.6	0.13	0.12	60.1
294	-0.72	3	13	0.11	-0.38	60.6
295	0.37	-3.4	5.4	0.41	0.83	34.7
296	-0.18	-0.4	8.6	0.11	-0.18	45
297	0.53	-1.8	4.9	0.19	0.4	27.6
298	-0.62	3	12.3	0.15	-0.13	68.2
299	0.02	-2.3	6.2	0.3	0.66	55.2
300	-1.76	3	10.5	0.29	0.27	94.7
301	-0.64	2	11.6	0.13	0.12	60.1
302	0.73	-1.8	5.2	0.19	0.44	22.8
303	-1.1	3	11.3	0.22	-0.36	103
304	0.02	-2.3	6.2	0.3	0.66	55.2
305	0.53	-1.8	4.9	0.19	0.4	27.6
306	-0.72	3	13	0.11	-0.38	60.6
307	0.73	-1.8	5.2	0.19	0.44	22.8
308	0.61	-2.5	5.2	0.29	0.82	25.5
309	0.53	-1.8	4.9	0.19	0.4	27.6
310	-1.1	3	11.3	0.22	-0.36	103
311	-0.69	0.2	10.5	0.18	-0.11	68.7
312	0.54	-1.5	5.9	0.14	0.27	23.7
313	0.37	-3.4	5.4	0.41	0.83	34.7

7.6 Physicochemical features for BID-18

This table includes the six physicochemical features calculated from the AAIndex database for the residues in BID-18 dataset except propensities which were calculated from [JT97; JT96]. AASA is Average Accessible Surface area.

Table 7.6.1: Physicochemical features for BID-18 dataset.

S.No.	Hydrophobicity	Hydrophilicity	Polarity	Polarizability	Propensities	AASA
314	0.61	-2.5	5.2	0.29	0.82	25.5
315	0.61	-2.5	5.2	0.29	0.82	25.5
316	0.61	-2.5	5.2	0.29	0.82	25.5
317	-1.1	3	11.3	0.22	-0.36	103
318	0.37	-3.4	5.4	0.41	0.83	34.7
319	-1.1	3	11.3	0.22	-0.36	103
320	0.16	0	9	0	-0.07	24.5
321	-1.76	3	10.5	0.29	0.27	94.7
322	0.73	-1.8	5.2	0.19	0.44	22.8
323	0.16	0	9	0	-0.07	24.5
324	-1.76	3	10.5	0.29	0.27	94.7
325	0.53	-1.8	4.9	0.19	0.4	27.6
326	0.16	0	9	0	-0.07	24.5
327	0.73	-1.8	5.2	0.19	0.44	22.8
328	0.54	-1.5	5.9	0.14	0.27	23.7
329	-0.62	3	12.3	0.15	-0.13	68.2
330	0.53	-1.8	4.9	0.19	0.4	27.6
331	-0.26	0.3	9.2	0.06	-0.33	42
332	-0.62	3	12.3	0.15	-0.13	68.2
333	-0.4	-0.5	10.4	0.23	0.41	50.7
334	-0.62	3	12.3	0.15	-0.13	68.2
335	-0.26	0.3	9.2	0.06	-0.33	42
336	0.53	-1.8	4.9	0.19	0.4	27.6
337	0.53	-1.8	4.9	0.19	0.4	27.6
338	0.25	-0.5	8.1	0.05	-0.17	27.8
339	0.53	-1.8	4.9	0.19	0.4	27.6
340	-0.72	3	13	0.11	-0.38	60.6
341	-1.76	3	10.5	0.29	0.27	94.7
342	0.54	-1.5	5.9	0.14	0.27	23.7
343	-0.72	3	13	0.11	-0.38	60.6
344	0.37	-3.4	5.4	0.41	0.83	34.7
345	0.02	-2.3	6.2	0.3	0.66	55.2
346	-0.69	0.2	10.5	0.18	-0.11	68.7
347	0.61	-2.5	5.2	0.29	0.82	25.5
348	0.54	-1.5	5.9	0.14	0.27	23.7
349	0.73	-1.8	5.2	0.19	0.44	22.8
350	0.61	-2.5	5.2	0.29	0.82	25.5
351	-1.1	3	11.3	0.22	-0.36	103
352	-0.07	0	8	0.13	-0.25	51.5
353	-0.69	0.2	10.5	0.18	-0.11	68.7

Continued on next page

Table 7.6.1 – continued from previous page

S.No.	Hydrophobicity	Hydrophilicity	Polarity	Polarizability	Propensities	AASA
354	-0.62	3	12.3	0.15	-0.13	68.2
355	0.53	-1.8	4.9	0.19	0.4	27.6
356	-1.76	3	10.5	0.29	0.27	94.7
357	-0.18	-0.4	8.6	0.11	-0.18	45
358	-1.76	3	10.5	0.29	0.27	94.7
359	0.02	-2.3	6.2	0.3	0.66	55.2
360	-0.62	3	12.3	0.15	-0.13	68.2
361	-1.1	3	11.3	0.22	-0.36	103
362	0.73	-1.8	5.2	0.19	0.44	22.8
363	0.26	-1.3	5.7	0.22	0.66	33.5
364	-1.1	3	11.3	0.22	-0.36	103
365	-1.1	3	11.3	0.22	-0.36	103
366	0.61	-2.5	5.2	0.29	0.82	25.5
367	0.26	-1.3	5.7	0.22	0.66	33.5
368	-0.18	-0.4	8.6	0.11	-0.18	45
369	0.61	-2.5	5.2	0.29	0.82	25.5
370	0.16	0	9	0	-0.07	24.5
371	-0.07	0	8	0.13	-0.25	51.5
372	0.53	-1.8	4.9	0.19	0.4	27.6
373	-0.18	-0.4	8.6	0.11	-0.18	45
374	0.37	-3.4	5.4	0.41	0.83	34.7
375	0.54	-1.5	5.9	0.14	0.27	23.7
376	0.37	-3.4	5.4	0.41	0.83	34.7
377	0.61	-2.5	5.2	0.29	0.82	25.5
378	-0.07	0	8	0.13	-0.25	51.5
379	-1.1	3	11.3	0.22	-0.36	103
380	-0.18	-0.4	8.6	0.11	-0.18	45
381	-0.64	2	11.6	0.13	0.12	60.1
382	-1.1	3	11.3	0.22	-0.36	103
383	0.73	-1.8	5.2	0.19	0.44	22.8
384	-0.62	3	12.3	0.15	-0.13	68.2
385	-0.69	0.2	10.5	0.18	-0.11	68.7
386	-1.1	3	11.3	0.22	-0.36	103
387	-0.26	0.3	9.2	0.06	-0.33	42
388	-0.72	3	13	0.11	-0.38	60.6
389	-0.26	0.3	9.2	0.06	-0.33	42
390	-1.1	3	11.3	0.22	-0.36	103
391	0.61	-2.5	5.2	0.29	0.82	25.5
392	-0.72	3	13	0.11	-0.38	60.6
393	0.02	-2.3	6.2	0.3	0.66	55.2
394	-0.62	3	12.3	0.15	-0.13	68.2

Continued on next page

Table 7.6.1 – continued from previous page

S.No.	Hydrophobicity	Hydrophilicity	Polarity	Polarizability	Propensities	AASA
395	0.53	-1.8	4.9	0.19	0.4	27.6
396	-1.76	3	10.5	0.29	0.27	94.7
397	0.61	-2.5	5.2	0.29	0.82	25.5
398	-0.18	-0.4	8.6	0.11	-0.18	45
399	0.54	-1.5	5.9	0.14	0.27	23.7
400	-1.76	3	10.5	0.29	0.27	94.7
401	-0.62	3	12.3	0.15	-0.13	68.2
402	-0.4	-0.5	10.4	0.23	0.41	50.7
403	-0.4	-0.5	10.4	0.23	0.41	50.7
404	-0.26	0.3	9.2	0.06	-0.33	42
405	0.16	0	9	0	-0.07	24.5
406	-0.72	3	13	0.11	-0.38	60.6
407	0.53	-1.8	4.9	0.19	0.4	27.6
408	-1.76	3	10.5	0.29	0.27	94.7
409	0.61	-2.5	5.2	0.29	0.82	25.5
410	0.73	-1.8	5.2	0.19	0.44	22.8
411	0.53	-1.8	4.9	0.19	0.4	27.6
412	-1.76	3	10.5	0.29	0.27	94.7
413	-0.72	3	13	0.11	-0.38	60.6
414	-0.4	-0.5	10.4	0.23	0.41	50.7
415	0.02	-2.3	6.2	0.3	0.66	55.2
416	0.02	-2.3	6.2	0.3	0.66	55.2
417	-0.62	3	12.3	0.15	-0.13	68.2
418	-1.76	3	10.5	0.29	0.27	94.7
419	-0.64	2	11.6	0.13	0.12	60.1
420	-1.76	3	10.5	0.29	0.27	94.7
421	-0.4	-0.5	10.4	0.23	0.41	50.7
422	-1.1	3	11.3	0.22	-0.36	103
423	-0.62	3	12.3	0.15	-0.13	68.2
424	0.61	-2.5	5.2	0.29	0.82	25.5
425	-1.1	3	11.3	0.22	-0.36	103
426	-1.1	3	11.3	0.22	-0.36	103
427	0.37	-3.4	5.4	0.41	0.83	34.7
428	-1.76	3	10.5	0.29	0.27	94.7
429	-1.1	3	11.3	0.22	-0.36	103
430	-1.76	3	10.5	0.29	0.27	94.7
431	-0.4	-0.5	10.4	0.23	0.41	50.7
432	-0.18	-0.4	8.6	0.11	-0.18	45
433	0.02	-2.3	6.2	0.3	0.66	55.2
434	0.04	-1	5.5	0.13	0.43	15.5
435	-0.72	3	13	0.11	-0.38	60.6

Continued on next page

Table 7.6.1 – continued from previous page

S.No.	Hydrophobicity	Hydrophilicity	Polarity	Polarizability	Propensities	AASA
436	-1.76	3	10.5	0.29	0.27	94.7
437	-1.76	3	10.5	0.29	0.27	94.7
438	0.61	-2.5	5.2	0.29	0.82	25.5
439	0.02	-2.3	6.2	0.3	0.66	55.2

7.7 PSSM for residues in HB-34

This table includes the twenty position specific scoring matrix (PSSM) based features calculated for the residues in HB-34 dataset.

Table 7.7.1: PSSM for residues in HB-34 dataset.

S.No.	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
1	0	0	2	-1	0	1	0	-2	2	-2	-1	-1	-1	0	-1	2	-1	1	-1	0
2	-4	1	2	3	-1	1	0	-3	2	0	-2	-1	-1	-2	-4	1	-3	4	1	-1
3	0	3	-1	0	0	1	-1	-2	3	-3	-2	-3	-2	-2	-1	3	-2	3	0	-2
4	-2	-1	0	-2	0	-1	3	-3	4	0	0	1	-2	0	-3	-1	1	1	2	-1
5	0	1	3	5	-1	-1	-2	-2	2	-1	-4	-2	-2	0	-7	1	-3	-1	1	-2
6	-2	0	2	3	-1	-1	0	0	1	-2	-3	1	-2	-1	-2	1	-1	4	0	-3
7	0	1	2	0	-3	2	0	-1	0	-4	-2	-1	0	-2	-3	3	-2	1	-1	-1
8	-2	0	-1	-3	-3	-2	2	-6	4	-1	-4	1	-4	-1	-8	2	2	0	3	1
9	-3	0	0	-2	-4	0	3	-4	2	0	-2	0	-2	1	-6	0	2	-1	1	2
10	0	1	1	0	1	0	-1	-1	0	-4	-2	-2	-1	-1	-2	3	0	3	0	-2
11	-3	-1	4	1	-4	1	0	2	3	-1	-2	0	-4	-4	-2	1	-1	-6	-1	-5
12	-2	1	1	2	-5	2	1	0	0	-3	-1	1	-2	0	0	0	-1	-1	2	-3
13	-2	2	1	0	0	0	1	-3	0	-1	-2	0	-2	-4	-3	1	3	-4	-3	1
14	-4	-5	-6	-6	-4	-4	-5	-6	-5	3	5	-5	3	1	-5	-5	-3	-4	-3	2
15	-1	1	0	-1	-3	2	-1	-3	-1	1	-2	2	0	-3	-2	0	1	-4	-2	1
16	0	-2	-1	3	-3	1	1	-2	0	-1	-2	-1	-2	-3	4	-1	-1	-4	-3	-1
17	-2	1	-1	0	-4	1	5	-1	-1	-4	-4	3	-3	-4	0	-1	-1	-4	-3	-3
18	1	-1	6	1	-3	0	-1	-1	0	-3	-3	-1	-2	-3	-2	0	0	-4	-2	-3
19	-1	-2	4	2	-2	-1	-1	1	-1	-2	-2	-1	-2	-2	-2	1	-1	6	-1	0
20	-1	-2	-1	-2	-2	-2	-2	1	0	1	-1	-2	-1	2	-2	0	-1	0	5	-1
21	-1	-2	-2	-3	-2	-1	-2	-3	3	1	1	-2	0	1	-2	-2	-1	0	5	1
22	-1	0	1	0	-2	1	2	2	0	-2	-2	1	-1	-2	-1	0	-1	-2	-1	-2
23	-1	-1	3	0	-2	-1	-1	-1	0	-1	-1	-1	-1	3	-1	0	-1	0	3	-1
24	-1	3	0	-1	-3	3	1	-2	-1	-3	-3	2	-1	-3	2	1	1	-3	-2	-2
25	-3	-3	-3	-4	-3	-3	-3	-4	3	-2	-1	-3	-2	3	-4	-3	-1	6	7	-1
26	-1	0	-1	2	-1	2	1	-3	2	-3	-2	0	-1	0	-4	1	2	1	2	-1

Continued on next page

Table 7.7.1 – continued from previous page

S.No.	**A**	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
27	-1	1	0	1	-3	0	1	-2	3	-1	-1	1	-2	-1	1	1	-1	-3	2	-1
28	-3	-2	3	1	-4	-2	-2	-1	1	-2	-3	-2	-2	0	-4	1	1	-2	6	-3
29	0	0	-1	0	-1	-3	-3	0	0	-3	-3	-2	0	1	-1	0	0	5	6	-2
30	-1	-1	2	0	-2	-1	-1	0	0	-2	-2	-1	-2	-1	2	2	0	1	2	-1
31	-3	-3	-3	-2	-4	-2	-2	2	-2	-2	-2	-3	-2	2	-4	1	-1	6	6	0
32	-3	-3	3	0	-5	-1	-2	0	1	-3	-2	-2	-2	-1	2	2	1	-4	4	-4
33	-1	-1	3	-1	-2	-1	-3	-1	1	-3	-4	-3	-4	1	-2	1	0	5	6	-3
34	-2	-1	-2	1	-2	0	-2	-3	-1	0	1	-1	2	2	-2	-2	-1	3	3	1
35	-1	1	-1	-1	-3	0	0	-2	-1	2	-1	4	0	-2	-2	-1	-1	-3	-2	0
36	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-3	-2	0
37	-2	-1	5	4	-3	0	0	-1	0	-3	-4	0	-3	-3	-2	0	0	-4	-3	-3
38	-1	2	0	-1	-3	1	1	-2	-1	-3	-3	5	-2	-3	-1	0	-1	-3	-2	-2
39	-1	-3	-4	-3	-1	-3	-3	-4	-4	5	2	-3	1	0	-3	-3	-1	-3	-1	3
40	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-3
41	-1	1	0	0	-3	6	2	-2	0	-3	-2	1	-1	-3	-1	0	-1	-2	-2	-2
42	1	5	-1	-2	-3	1	0	-2	-1	-3	-2	3	-1	-3	-2	0	-1	-3	-2	-2
43	0	-2	-1	-2	-1	-1	-1	-2	-2	0	2	-1	1	-1	-2	2	0	-2	-1	0
44	-1	3	0	0	-3	1	3	3	-1	-4	-3	1	-2	-3	-2	0	-1	-3	-2	-3
45	-2	-1	5	4	-3	0	0	-1	0	-3	-4	0	-3	-3	-2	0	0	-4	-3	-3
46	-3	-3	-4	-5	-3	-2	-3	-3	-3	-3	-2	-3	-2	1	-4	-3	-3	11	2	-3
47	-1	2	0	-1	-3	1	1	-2	-1	-3	-3	5	-2	-3	-1	0	-1	-3	-2	-2
48	0	-1	3	0	-2	0	0	-1	0	-3	-3	0	-2	-1	-2	3	1	-1	2	-2
49	-1	2	-1	-1	-3	1	0	-2	-1	0	-1	4	-1	-2	-1	-1	-1	-3	-2	-1
50	0	-2	-1	-2	-2	-1	-1	1	0	-1	0	-1	-1	1	-2	2	0	0	4	-1
51	-2	-2	1	6	-4	0	2	-1	-1	-3	-4	-1	-3	-4	-2	0	-1	-5	-3	-3
52	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-3	-2	0
53	-1	-2	1	6	-3	0	1	-1	-1	-3	-4	-1	-3	-4	-1	1	0	-4	-3	-3

Continued on next page

Table 7.7.1 – continued from previous page

S.No.	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
54	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-3
55	-1	-1	-2	-1	-2	0	2	-3	-2	1	3	-1	3	0	-2	-1	-1	-2	-1	0
56	-2	-3	-4	-4	-2	-3	-3	-4	-3	2	3	-3	2	2	-3	-3	-1	-1	0	1
57	-1	-2	-3	-3	-2	-2	-2	-3	1	1	0	-2	0	2	-3	-2	-1	1	6	2
58	-2	0	-2	-2	-3	0	-2	-1	-2	0	3	0	0	1	-1	-2	-1	2	3	-1
59	-1	0	-4	-4	-3	-3	-2	-5	-4	0	4	-1	1	2	-4	-1	0	-3	-1	0
60	-2	1	0	-1	-3	1	0	-1	3	-2	1	2	-1	1	-1	-1	-1	-2	3	-2
61	-1	3	-2	1	-4	2	4	0	-2	-4	-4	1	-3	-4	0	1	0	-4	-3	-2
62	-1	-1	-2	-1	-3	1	-1	-1	0	0	-1	0	-1	0	-3	0	1	7	1	0
63	-1	1	-1	0	-2	-1	1	-1	-1	-1	-1	1	0	-2	0	1	1	-2	1	0
64	-1	-2	-1	1	0	-1	1	-2	-2	1	0	0	-1	1	0	0	3	-3	-2	0
65	-1	-1	1	0	1	1	-2	-1	-2	-2	0	-1	0	0	-1	4	0	-3	-2	-1
66	-1	-3	-2	-2	-2	-2	-3	-3	0	3	2	-2	0	2	-1	1	-1	-3	-1	2
67	-3	-3	-3	0	-4	-1	1	-2	-2	-3	-2	-2	-1	4	-2	-2	0	8	4	-3
68	-1	0	-2	0	-3	2	0	-2	-2	2	-1	0	-1	-1	-1	0	2	-3	-1	2
69	-1	-1	0	-2	-3	0	-1	-2	1	1	-2	-1	-1	-1	4	-1	1	1	-2	1
70	0	-1	-1	-2	2	2	0	0	1	-4	-2	-2	-1	-2	4	2	0	-5	-4	-4
71	-2	0	-1	-1	-3	2	1	-2	0	1	1	1	-1	-2	2	0	-1	-4	-1	0
72	1	-1	2	2	-3	1	-1	-3	3	-3	-2	-1	-3	0	-1	2	2	-4	-1	-2
73	-1	0	2	1	-2	2	0	-3	-1	-2	0	-1	0	0	-3	-2	3	-4	-2	0
74	-1	0	1	3	-4	-1	2	-2	1	-1	-3	0	-3	-3	1	0	0	1	2	-1
75	-1	-1	-1	-1	-2	0	-1	-1	0	1	0	-1	2	2	0	0	0	1	-1	1
76	0	2	0	1	-3	1	0	-2	1	-3	-3	2	-2	-3	3	0	0	-3	-1	-3
77	0	0	1	1	-1	0	0	2	1	-2	-2	1	-1	-3	0	2	0	-3	-1	-2
78	-1	-1	0	-1	2	0	-1	1	1	-1	0	-1	-1	0	-1	0	-1	6	2	-1
79	-1	-1	-1	-1	1	-1	-1	-2	-1	2	1	0	0	-2	0	0	2	-3	-2	1
80	0	1	2	1	-3	1	1	-2	0	-2	0	0	0	-2	-1	0	0	-3	0	-2

Continued on next page

Table 7.7.1 – continued from previous page

S.No.	**A**	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
81	0	-1	0	0	-3	1	-1	0	4	-1	-3	1	-3	1	0	2	-1	-3	1	-2
82	-2	-2	-2	-3	-3	-2	-3	2	1	-2	-2	-2	-2	2	-3	-2	-2	1	7	-2
83	-2	-2	-2	1	-2	-2	-2	-3	-1	0	3	-2	0	0	3	-2	-1	-1	2	0
84	-2	-2	-2	1	-3	-1	1	-3	-1	-2	-1	-2	-1	5	3	-2	-2	0	3	-2
85	-1	1	2	4	-3	2	2	-2	3	-3	-3	0	-2	-3	-2	0	-1	-3	-1	-3
86	1	-2	-3	-3	-1	-2	-2	-2	-3	2	2	-2	3	-1	-2	-1	-1	-3	-1	3
87	-1	-2	1	5	-4	0	2	2	-1	-4	-4	-1	-3	-4	-2	-1	-1	-4	-3	-3
88	-2	-2	-3	-4	-3	-2	-3	-4	1	-1	-1	-3	-1	5	-4	-2	-2	2	7	-2
89	-1	-2	-2	-3	-2	-1	-2	-3	2	1	1	-2	4	2	-3	-2	-1	0	4	1
90	-2	-2	-3	-4	-3	-2	-3	-4	1	-1	0	-3	-1	4	-4	-2	-2	1	7	-1
91	0	-2	2	0	-1	0	1	-2	-1	-1	-2	0	-2	-4	5	0	-2	-4	0	1
92	-1	1	1	0	-3	1	1	-3	3	-2	-2	-1	-2	-1	0	2	2	-4	-1	-1
93	0	2	0	-2	-3	2	1	-3	1	0	-1	0	-2	1	-3	1	2	0	2	-2
94	-2	2	-1	-1	-3	0	0	-2	0	-2	1	3	2	0	-3	0	0	-2	2	-2
95	0	-2	3	4	-3	0	1	1	-2	-3	-3	-1	-3	-2	-2	1	-1	-3	-1	-2
96	0	1	0	1	-3	4	1	0	-1	-2	0	-1	0	-2	-1	0	0	0	-2	-2
97	-1	-1	-1	-1	-2	1	0	-2	1	0	1	1	-1	-2	2	0	1	-3	-2	1
98	-1	0	1	1	-1	2	1	0	1	-1	-1	0	-1	0	1	0	0	-2	0	-1
99	-1	1	0	0	-2	0	0	0	-1	-2	-1	1	-1	0	-1	1	0	0	2	-1
100	-1	-1	-2	-1	-2	-2	-1	-2	-2	1	2	-1	2	0	0	-1	-1	5	2	0
101	-1	-1	0	0	-1	0	-1	-2	2	1	0	-1	-1	0	0	1	1	-2	1	0
102	0	1	2	1	-3	0	1	-1	2	-2	-2	1	-2	0	-1	1	0	-2	0	-2
103	-1	1	0	1	1	0	1	-1	1	0	-1	1	-1	0	-1	1	0	-2	0	-1
104	0	0	3	0	-2	-1	1	-1	-1	-2	-3	0	-1	-2	2	2	0	-3	0	-2
105	-1	0	0	3	-2	0	1	0	3	-2	-2	0	-2	0	-2	0	1	-2	-2	-2
106	0	1	1	0	-2	3	0	-1	1	-2	-2	1	-2	-2	1	1	-1	-2	2	-1
107	-1	0	-1	0	-3	2	3	-3	0	-1	-2	0	1	1	-2	1	-1	0	-2	0

Continued on next page

Table 7.7.1 – continued from previous page

S.No.	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
108	-4	-1	0	1	-6	5	3	-2	7	-6	-5	-1	-4	-5	-4	-2	-2	0	-3	-5
109	-4	1	1	-4	-2	-3	-3	-5	9	-5	-5	-2	-2	0	-5	1	-1	-4	4	-3
110	-1	1	3	1	-3	2	0	-2	1	0	0	0	0	-2	-3	1	0	-4	-2	-2
111	2	-1	-1	1	-3	1	1	-1	-2	0	-2	0	0	-1	-2	2	2	-4	-2	-1
112	-3	1	-2	3	-5	2	4	-4	0	-1	-4	2	-2	-5	-3	-1	-1	-5	-4	-4
113	-1	3	-1	-2	-3	3	0	-2	3	-2	0	0	0	0	-2	0	0	-2	2	-1
114	-1	-1	0	-2	-2	-1	-2	-3	-2	2	2	0	1	2	-2	-1	0	-2	-1	2
115	-1	1	0	-2	-3	2	-1	-3	2	1	0	1	1	-3	-1	0	1	-3	-2	0
116	0	-2	2	0	0	-1	-1	0	-2	-2	-2	0	1	-2	1	2	2	-3	0	-1
117	0	-1	2	3	-3	-1	2	-2	-2	-3	-3	1	-3	-2	-3	3	0	-4	1	-2
118	-1	-2	2	-1	-3	0	4	-2	-1	1	-3	0	-1	-3	0	1	2	-4	-3	0
119	-3	-3	4	5	-4	-2	0	1	-3	-5	-5	-2	-4	-5	-1	2	0	-5	-4	-4
120	-2	-1	0	3	-5	0	5	-1	4	-5	-4	-1	-4	-5	-3	1	0	-5	-3	-4
121	2	2	0	-3	-2	0	-2	-3	4	-4	-4	-1	-1	1	-4	3	0	-3	4	-3
122	-3	5	-2	0	-5	-1	-2	-2	1	-5	-2	4	-3	-2	-4	-3	-3	-4	1	-3
123	-4	2	-1	0	-6	2	-1	-5	9	-6	-5	1	-4	-2	-5	-1	-4	-5	0	-6
124	-4	0	-2	-4	-6	-2	-2	-4	0	-5	-5	7	0	-2	-4	-3	0	-5	-1	-3
125	-2	-3	2	0	-1	-3	-3	-3	1	0	-1	-3	2	1	-3	2	5	-4	-3	-2
126	-3	-5	-3	-6	-5	-4	-5	-6	0	-3	-1	-3	-3	5	-6	-3	-4	-1	8	-4
127	-3	4	0	-3	-5	1	0	-4	-3	-5	-5	6	-3	-5	-4	-1	1	-5	-4	-4
128	-2	-4	-5	-5	-3	-4	-3	-5	-1	3	3	-4	2	1	-5	-4	-1	-4	-1	4
129	-4	7	-1	-3	-6	1	-3	-5	0	-5	-2	1	-4	-5	-5	-3	-4	-5	-1	-5
130	-2	-2	1	0	-4	-2	-1	-3	1	-3	-3	-2	-3	1	-4	1	0	-1	7	-3
131	0	1	2	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	1	-1	1	-1	2	4	-1
132	-1	-1	0	0	-2	-1	0	-1	1	0	-1	0	-1	-1	2	1	0	2	2	-1
133	0	-1	3	2	-2	0	0	1	0	-3	-2	0	-2	-1	-1	1	0	0	-1	-2
134	1	-2	0	3	-3	-1	0	4	-3	-3	-2	-2	-3	-3	-3	1	-1	-4	-2	-1

Continued on next page

Table 7.7.1 – continued from previous page

S.No.	**A**	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
135	1	-2	1	2	-3	0	0	1	-1	-3	-3	-1	-3	-3	1	2	0	-4	-1	-2
136	-1	-1	4	0	-3	0	0	-2	2	-2	-2	0	-2	-1	-2	0	0	-1	4	-1
137	3	-1	3	-2	-3	-1	-1	1	-2	-3	-3	-2	-3	-4	3	1	0	-4	-3	-2
138	0	-2	0	-2	-4	-3	-2	1	1	-1	-2	-3	1	5	3	0	-1	2	0	-2
139	-2	-6	0	-5	-6	-5	-5	7	-5	-6	-6	-5	-6	-6	-6	-4	-5	-5	1	-6
140	0	-2	1	-1	5	-1	-1	0	-2	0	-1	-1	-1	-2	-2	1	2	-2	-2	0
141	-1	-1	1	-1	-4	3	-1	3	0	-3	-3	-1	-2	-3	-2	2	2	-5	-2	-3
142	-1	-1	1	-1	-3	-2	-1	0	3	-2	-1	-2	-3	3	0	2	0	-3	1	-1
143	-1	0	1	1	-3	0	0	1	-1	0	-1	-1	-2	0	-1	1	0	2	2	-1
144	-2	-2	1	-1	-2	-2	-1	0	0	-1	-1	-2	-3	2	-3	1	0	6	4	0
145	0	1	3	1	-3	-1	-2	1	-1	-2	-2	-2	-2	-1	-2	0	1	1	3	-1
146	-1	-1	2	0	-3	0	0	1	-1	-2	-2	0	-3	-1	1	3	0	-3	1	-2
147	-1	-1	0	-2	-4	0	0	-1	-1	-3	-1	0	0	-2	4	1	1	-1	0	-2
148	-2	-2	1	6	-4	0	2	-1	-1	-3	-4	-1	-3	-4	-2	0	-1	-4	-3	-3
149	-1	2	0	-1	-3	1	1	-2	-1	-3	-3	5	-1	-3	-1	0	-1	-3	-2	-2
150	-2	-3	-3	-4	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
151	-1	-1	-2	-2	-3	-2	-2	-3	-3	-3	-3	-1	-3	-4	8	-1	-2	-4	-3	-3
152	0	-1	-1	-1	-2	1	1	-2	-1	0	0	-1	0	-2	4	-1	0	-2	-2	1
153	1	-1	-1	-1	-2	1	0	-1	3	-2	-2	-1	-2	-2	5	0	-1	-3	-1	-2
154	1	-1	0	-1	-1	-1	-1	4	-1	-2	-2	-1	-1	-2	2	0	-1	-1	-1	-1
155	0	-2	-1	-1	-2	-1	-1	-1	-2	-2	-2	-1	-2	-2	7	-1	-1	-2	-2	-2
156	1	-2	-2	-1	-2	-1	-1	2	-2	-2	-3	-1	-2	-3	6	0	-1	-3	-2	-2
157	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
158	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
159	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	10	2	-3
160	0	0	0	-1	2	1	2	0	1	-2	0	0	-1	-2	0	0	0	-3	0	-1
161	-1	1	1	0	2	0	1	-1	0	0	-1	0	0	-1	0	0	0	0	0	0

Continued on next page

Table 7.7.1 – continued from previous page

S.No.	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
162	-2	2	-1	-2	-4	2	1	-3	3	-1	-2	1	-2	-1	-3	0	2	-4	1	0
163	-2	2	-1	-1	-5	2	-1	-3	1	-3	0	0	-4	-2	5	-1	-2	-1	1	-2
164	-1	1	-1	0	-3	2	2	-1	0	-2	-2	1	0	-1	-3	1	2	1	0	-1
165	-2	0	-2	-2	9	1	0	-2	0	-2	-3	-2	-1	-1	-2	-3	-3	-6	-1	-1
166	-2	-1	-1	-1	2	0	0	-2	2	-1	-1	-2	0	-1	3	0	1	2	1	-1
167	-3	-2	1	-1	7	0	1	-1	1	-1	-2	-1	-1	-1	1	-1	-1	-7	1	-1
168	-2	1	2	1	2	1	1	0	1	-2	-1	0	-1	1	-1	-1	-2	1	0	-3
169	-1	-1	5	1	-2	0	0	0	0	-3	-3	0	-2	-3	-2	2	0	-4	-2	-3
170	-2	2	-2	-2	-3	-1	-1	-3	0	-1	-1	1	-1	4	-3	-1	-2	1	4	-1
171	0	0	1	-1	-2	2	0	-1	3	-1	1	-1	1	0	-2	0	-1	-2	1	-1
172	-2	-2	0	1	-3	-2	-2	-3	1	-2	-2	-2	-2	5	-3	-1	-1	0	6	-2
173	-1	4	-1	-1	-3	3	1	0	1	-3	-1	2	-2	-3	-2	-1	-2	1	-1	-2
174	-4	-5	-5	-6	-4	-4	-5	-4	-4	-4	-4	-5	-3	0	-5	-5	-4	12	0	-5
175	-1	3	0	1	-3	0	0	-3	0	-2	-2	2	-2	-2	2	-1	-1	-3	1	1
176	1	0	-2	-1	-2	-1	0	0	2	1	1	-1	0	-1	-2	-1	-1	-3	-1	1
177	0	-1	-1	0	-2	-1	0	-2	-2	-1	0	1	-1	-3	-2	1	5	-3	-2	-1
178	-2	1	-2	-2	-5	1	-1	-3	-2	-4	-3	7	2	-4	-3	-2	-2	-4	-3	-3
179	-2	4	-1	-2	-4	1	-1	-3	0	-4	-3	6	-3	-4	-3	0	0	-4	-3	-3
180	-2	-4	-5	-5	-3	-4	-5	-5	-5	6	0	-4	0	-2	-4	-4	-2	-4	-3	4
181	0	2	-1	-1	-2	2	0	-1	-1	-3	-2	3	-2	-3	-1	2	0	-3	-2	-2
182	-2	2	1	4	-4	1	2	1	-1	-3	-2	-1	-3	-4	-2	0	-2	-4	-3	-3
183	-1	-2	-2	-3	-2	-2	-2	-3	-3	4	1	0	1	-1	-3	0	-1	-3	-2	2
184	0	-2	2	-1	-1	-1	-1	-2	-2	0	-1	-1	-1	-2	2	1	4	-3	-2	0
185	-1	4	-1	-1	-3	3	1	-2	0	-3	-3	3	-1	-3	1	-1	-1	-3	-2	-3
186	-1	-2	1	-1	-1	-1	-1	-2	-2	1	-1	-1	1	-2	-2	1	5	-3	-2	0
187	0	2	0	-1	-2	2	0	3	-1	-2	-2	0	-1	-2	-1	0	0	-1	-1	-1
188	-1	4	0	-1	-3	1	0	0	-1	-3	-2	3	-2	-3	-1	0	-1	-3	-2	-2

Continued on next page

Table 7.7.1 – continued from previous page

S.No.	**A**	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
189	0	4	0	-1	-3	0	0	1	-1	-3	-3	3	-2	-3	-2	1	-1	-3	-2	-2
190	0	2	-1	-2	3	0	-1	4	-2	-3	-3	2	-2	-3	-2	-1	-1	-3	-3	-3
191	-1	6	-1	-2	-4	1	0	-3	-1	-3	-3	2	-2	-3	-2	0	-1	-3	-2	-3
192	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
193	0	-1	0	-1	-2	0	-1	-1	-2	-2	-2	1	-2	-3	-1	4	4	-3	-2	-1
194	2	-1	0	-1	-1	-1	0	-1	-2	-3	-3	-1	-2	-3	2	4	0	-3	-2	-2
195	1	5	-2	-2	-3	0	0	-3	-2	-4	-3	1	1	-4	2	2	-2	3	-3	-3
196	-1	2	0	1	-4	-1	-2	-3	2	-4	-4	4	-3	-3	-4	-2	-1	8	-2	-4
197	-3	-1	-2	-3	-5	7	0	-4	0	-5	-2	3	-3	-5	-4	0	-2	-5	-4	-4
198	-5	-3	-3	-2	-6	0	-3	-5	11	-6	-6	-4	-5	-4	-5	-4	-5	-5	1	-6
199	0	4	0	0	-2	1	-1	-2	-2	-2	-2	3	-1	-2	-3	1	-1	-4	-3	-1
200	-1	4	0	-1	0	1	-1	-1	-1	0	-2	2	0	-3	-1	0	-1	-4	-3	0
201	-2	-2	6	1	-4	0	0	-1	-1	-4	-3	2	-3	-3	-3	-1	-2	-4	1	-4
202	-2	1	-2	-1	-3	-1	-2	-3	0	1	-3	2	-1	-4	-2	1	5	-4	-3	0
203	-1	3	-3	-4	-4	-2	-3	-4	2	2	-2	-2	-2	0	2	-3	0	2	6	-2
204	-3	1	2	4	-5	-1	5	-3	-2	-5	-3	-1	-4	-5	-3	-2	-3	0	-4	-3
205	-5	-3	0	-1	-6	-2	-3	-5	11	-6	-6	-4	-5	-4	-5	-4	-5	-5	-1	-6
206	-3	-3	2	0	-4	-2	-2	-3	1	-1	-3	-2	-1	1	-4	1	1	-2	7	-2
207	-1	1	1	0	-3	-1	-1	-2	0	-1	-2	-1	-1	1	1	1	-1	2	5	-2
208	0	-1	2	3	-3	1	1	1	0	-2	-2	-1	-2	-2	-2	2	0	-3	-2	-2
209	1	-1	2	2	-3	1	1	1	-2	-2	-3	-1	-2	-1	-1	2	1	-3	-1	-1
210	-1	0	3	1	-3	1	1	-1	2	-1	-2	1	-1	-2	0	0	0	-3	3	-2
211	-1	-1	-1	1	-4	1	-1	1	-2	-2	-1	-2	-1	1	-3	1	-1	4	4	0
212	0	-2	1	2	-3	-2	0	2	-1	-2	-3	-1	-3	-2	-1	2	1	-3	2	-3
213	-1	-1	1	-2	-4	-2	-2	1	-1	-2	-1	-3	-1	1	1	1	0	2	4	-2
214	-2	-4	0	-2	-2	-2	-1	-2	1	-3	0	-3	-2	1	4	0	0	2	5	-3
215	-1	-1	0	-1	-3	-2	-2	2	0	0	-4	-1	-3	-4	-2	4	3	-4	-3	-2

Continued on next page

Table 7.7.1 – continued from previous page

S.No.	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
216	0	-1	0	-1	-1	-3	-3	0	1	-3	-2	-2	-3	1	-4	0	0	5	6	-2
217	1	0	2	1	-3	0	0	-1	0	-2	-2	-1	-1	0	-1	1	0	1	3	-1
218	0	-2	0	1	-3	-1	1	2	-1	-2	-2	-2	-2	-1	1	2	0	-3	2	-1
219	-1	-1	2	-2	-3	-1	-1	0	-1	-1	-1	-2	-2	1	1	1	0	3	4	-2
220	-3	-4	-4	-4	-4	-4	-2	-4	2	1	-2	-4	-1	0	-3	-2	3	-3	5	4
221	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
222	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
223	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
224	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
225	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
226	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
227	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
228	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
229	-3	-1	-2	-3	-4	-2	-3	-1	-1	-2	-2	-1	-4	2	-4	1	-1	2	8	-3
230	-1	0	-1	0	-3	-1	0	1	-1	0	-1	-1	-1	1	-1	1	-1	2	3	-1
231	-1	-1	3	2	-3	-2	-2	1	-1	-3	-2	-2	-3	-1	-2	1	0	1	3	-3
232	-2	-2	-4	-4	-1	-2	-3	-4	-3	2	4	-3	2	0	-3	-3	-1	-2	-1	1
233	-2	-3	-3	-4	-3	-4	-4	-3	-1	0	0	-3	0	7	-4	-3	-2	1	3	-1
234	-2	-2	1	6	-4	0	2	-1	-1	-3	-4	-1	-3	-4	-2	0	-1	-5	-3	-3
235	1	-1	3	0	-1	0	0	0	0	-3	-3	0	-2	-3	-1	4	1	-3	-2	-2
236	-1	-3	-3	-3	-1	-3	-3	-4	-4	4	1	-3	1	0	-3	-2	-1	-3	-1	3
237	3	-1	0	-1	-1	0	0	0	-1	-2	-2	0	-1	-3	-1	3	1	-3	-2	-1
238	-1	0	0	2	-4	2	5	-2	0	-4	-3	1	-2	-4	-1	0	-1	-3	-2	-3
239	-1	-3	-3	-3	-1	-3	-3	-4	-3	3	1	-3	1	-1	-3	-2	0	-3	-1	4
240	-1	-1	0	4	-4	1	4	-2	-1	-3	-3	0	-3	-4	-1	0	-1	-4	-3	-3
241	-1	0	0	2	-4	2	5	-2	0	-4	-3	1	-2	-4	-1	0	-1	-3	-2	-3
242	-1	1	-1	-1	-3	2	1	-2	-1	-2	-1	4	-1	1	-2	-1	-1	-2	-1	-2

Continued on next page

Table 7.7.1 – continued from previous page

S.No.	**A**	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
243	-2	-2	-2	-2	-3	-1	0	-3	0	-2	-2	-2	-2	1	2	-2	-2	1	7	-2
244	0	2	0	-1	-2	0	0	0	2	-2	-2	2	-1	-2	-1	1	0	-2	-1	-2
245	0	-1	1	1	-1	1	0	-1	-1	-1	0	0	-1	-1	-1	2	1	-2	-1	-1
246	-1	0	-1	1	-2	-1	0	2	2	-2	-2	-1	-2	-1	-2	1	-1	7	0	-2
247	-1	0	2	1	-2	1	1	-2	-1	0	-2	2	-1	-3	-2	1	0	-3	-2	-1
248	-2	-2	1	-3	-3	-2	-2	-3	0	-2	0	-2	-1	1	-3	-2	0	6	6	-2
249	-2	7	-1	-2	-4	0	-1	-3	-1	-4	-3	3	-2	-4	-3	-2	-2	-4	-3	-3
250	-1	3	-1	-1	-3	1	3	-2	-1	-2	-1	2	2	-2	-2	-1	1	-3	-2	-2
251	-2	-2	-2	-3	-3	-2	-2	-3	2	-2	-1	-2	-1	3	-3	-2	-2	2	7	-1
252	-2	-2	1	6	-4	0	2	-1	-1	-3	-4	-1	-3	-4	-2	0	-1	-5	-3	-3
253	-2	-2	1	6	-4	0	2	-1	-1	-3	-4	-1	-3	-4	-2	0	-1	-5	-3	-3
254	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-3	-2	0
255	3	-2	-2	0	-3	0	5	-2	-2	-4	-4	-1	-3	-4	-2	-1	-2	-4	-3	-3
256	0	-1	-1	-1	-2	-1	-1	3	-2	-1	-2	1	-1	-2	-2	0	2	-3	-2	0
257	2	0	0	2	-1	0	0	-1	-1	-2	-2	2	-2	-3	-1	2	0	-3	-2	-1
258	1	-1	0	0	-1	0	2	-1	-1	-2	-2	0	-1	-3	-1	2	3	-3	-2	-1
259	-1	-2	0	3	-2	-1	0	-2	3	0	1	-1	0	-1	-2	-1	-1	-3	-1	1
260	-1	-2	1	2	-2	-1	-1	-2	-2	2	1	-2	0	-1	-2	-1	-1	-3	-2	2
261	0	-2	-2	-2	-1	-2	-2	-2	-2	2	2	-2	1	-1	-2	-1	0	-2	-1	3
262	-1	1	-1	0	-2	1	3	-2	-1	-1	0	0	0	-2	-2	0	2	-3	-2	-1
263	-1	0	-1	0	-2	1	3	-2	-1	1	-1	2	-1	-2	-2	-1	-1	-3	-2	1
264	-2	0	0	-1	-3	0	0	-2	8	-4	-3	-1	-2	-1	-2	-1	-2	-3	2	-3
265	0	-1	1	3	-2	0	0	-1	-1	-3	-3	0	-2	-3	-1	3	1	-3	-2	-2
266	1	-1	0	0	-1	0	0	-1	-1	-2	-2	0	-1	-2	-1	4	2	-3	-2	-1
267	-2	-2	1	6	-4	0	1	-1	-1	-3	-4	-1	-3	-4	-2	0	-1	-4	-3	-3
268	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-3	-1	-3	-1	2
269	-2	-2	-2	-3	-3	-2	-2	-3	2	-2	-1	-2	-1	3	-3	-2	-2	2	7	-1

Continued on next page

Table 7.7.1 – continued from previous page

S.No.	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
270	-2	-2	-2	-3	-3	-2	-2	-3	2	-2	-1	-2	-1	3	-3	-2	-2	2	7	-1
271	-1	2	3	1	-4	0	3	2	-1	-4	-4	1	-2	-4	-3	1	0	-4	-3	-3
272	0	-2	-2	-2	-1	-2	-2	-2	-2	2	2	-2	1	-1	-2	-1	0	-2	-1	3
273	-1	0	-1	0	-2	1	3	-2	-1	1	-1	2	-1	-2	-2	-1	-1	-3	-2	1
274	1	-1	0	0	-1	0	0	-1	-1	-2	-2	0	-1	-2	-1	4	2	-3	-2	-1
275	-1	-2	-2	-2	-3	-1	-1	-2	-2	-3	-3	-1	-3	-4	8	-1	-1	-4	-3	-3
276	-2	-2	0	-2	-3	-1	-1	-3	-3	0	4	-2	2	0	2	0	-1	-4	-3	-1
277	-2	-3	-3	-4	-3	-4	-4	-3	-1	0	0	-3	0	7	-4	-3	-2	1	3	-1
278	-1	-2	-1	-2	-2	-2	-2	-3	1	-2	-2	-2	-2	-3	2	0	6	-3	-2	-1
279	-3	-2	2	6	-4	-1	2	-2	-2	-4	-4	-1	-4	-4	-2	-1	-2	-5	-4	-4
280	-3	-3	-3	-4	-3	-3	-3	-4	-1	0	2	-3	0	5	-4	-3	-2	0	5	-1
281	-2	7	-1	-3	-5	0	-1	-3	-1	-4	-3	1	-2	-4	-3	-2	-2	-4	-3	-4
282	-2	6	-2	-3	-4	0	-1	-3	-1	-1	-2	1	-2	2	-3	-2	-2	5	-1	-2
283	-1	0	-3	-2	-3	1	0	-3	0	-1	-3	0	-3	-1	3	1	3	-4	-2	1
284	-2	4	1	-2	-4	0	-1	0	0	-4	-1	3	1	-1	-4	-1	2	-4	0	-3
285	-1	0	1	-3	-4	-1	-1	-2	2	-1	-1	-1	1	2	-1	2	-2	3	4	-1
286	-1	1	-3	-3	-4	1	-1	-2	1	-1	-1	0	-3	0	5	-1	1	-4	-2	-1
287	-2	-2	-2	-2	-4	-1	-3	-5	-1	4	0	-1	4	0	-3	-2	1	3	-2	2
288	-2	2	1	-1	-5	2	2	4	1	-5	-2	0	-1	-4	-2	-2	-3	-4	0	-4
289	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0
290	-1	0	0	1	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
291	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	4	-1	-3	-1	0	-1	-3	-2	-2
292	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	4	-1	-3	-1	0	-1	-3	-2	-2
293	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
294	-2	-2	1	6	-3	0	1	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
295	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	10	2	-3
296	-1	0	-3	-3	-4	-3	-3	-4	-1	-3	-3	-1	0	-4	5	1	5	-5	-4	-1

Continued on next page

Table 7.7.1 – continued from previous page

S.No.	**A**	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
297	-1	4	-2	-2	-3	1	-2	-1	0	-1	1	1	3	-3	0	-1	-1	-4	-3	-1
298	-2	-1	0	2	-4	1	4	-3	2	0	-1	0	0	-3	-3	0	0	-4	-1	0
299	-3	-2	-1	-2	-4	0	-2	-4	1	-3	-1	-2	1	2	-2	-1	-3	3	7	-1
300	-1	2	3	2	-3	0	0	-3	-2	-1	-1	1	3	1	-3	-1	-1	-3	-1	-1
301	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
302	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
303	-1	4	0	-1	-3	1	0	-2	-1	-3	-2	4	-1	-3	-1	0	-1	-3	-2	-2
304	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
305	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
306	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-4	-1	0	-1	-4	-3	-3
307	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
308	-2	-3	-3	-4	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
309	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
310	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
311	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
312	0	-3	-3	-3	-1	-2	-3	-3	-3	3	1	-3	1	-1	-3	-2	0	-3	-1	4
313	-3	-3	-4	-5	-3	-2	-3	-3	-3	-3	-2	-3	-2	1	-4	-3	-3	12	2	-3

7.8 PSSM for BID-18 residues

This table includes the twenty position specific scoring matrix (PSSM) based features calculated for the residues in BID-18 dataset.

Table 7.8.1: PSSM for residues in BID-18 dataset.

S.No.	**A**	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
1	-2	-4	-5	-7	0	-6	-7	-7	0	2	4	-6	2	4	-7	-5	-5	4	2	0
2	-3	-4	-7	-7	-1	-5	-6	-7	-3	2	2	-6	3	7	-7	-5	-6	-2	3	-1
3	-2	-6	-4	-4	-1	-6	-5	-7	-2	2	2	-5	0	7	-7	-4	-5	-4	5	-2
4	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
5	-3	-3	-4	-4	-2	-2	-3	-3	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3
6	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
7	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-3	-3	-3
8	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
9	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
10	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-3	-3	-3
11	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
12	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
13	-2	0	1	1	-5	-1	-2	4	2	-3	-4	-1	-3	-3	-1	2	1	2	-3	-4
14	-1	2	1	-1	-4	2	0	-3	2	-1	-2	1	0	-2	0	1	1	1	0	0
15	-3	2	-3	-5	-3	0	-3	-4	0	2	0	0	1	1	-5	-2	0	5	3	2
16	2	1	-2	0	-3	0	2	-2	0	0	-1	0	1	-3	-3	0	0	-4	-2	0
17	-3	0	-4	-3	-4	1	-1	-3	-1	2	3	0	0	-1	-2	0	-1	0	-2	0
18	-2	2	2	0	-4	-1	-1	-2	0	-4	-2	1	-2	-1	-4	3	2	-3	0	-2
19	-1	1	0	-1	-3	1	1	-2	0	-1	-1	1	-1	-2	0	0	0	-2	0	2
20	-1	1	0	0	-4	1	0	-2	2	-1	0	1	-1	-1	2	0	0	2	1	-2
21	0	0	-1	-1	-1	2	3	1	0	-2	-2	1	-2	-3	-2	0	-1	0	-1	-1
22	-2	2	-2	-2	-4	0	0	-3	1	1	0	1	0	0	-2	-1	2	1	-1	2
23	-3	-2	-1	-1	-4	1	-2	-5	-1	2	2	-3	1	-3	-4	-1	5	-5	-2	0
24	-1	0	2	4	-5	0	0	-3	-1	-2	-2	-1	-3	-2	1	1	1	-5	-1	-2
25	4	-2	-2	-2	0	-1	-1	0	-2	-1	-2	-1	-1	-2	-1	1	0	-3	-2	0
26	-2	-2	-4	-4	-1	-2	-3	-4	-3	2	4	-3	2	0	-3	-3	-1	-2	-1	1

Continued on next page

Table 7.8.1 – continued from previous page

S.No.	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
27	-2	-2	1	6	-4	0	2	-1	-1	-3	-4	-1	-3	-4	-2	0	-1	-5	-3	-3
28	-2	6	0	-2	-4	1	0	-3	0	-3	-2	2	-2	-3	-2	-1	-1	-3	-2	-3
29	0	-3	-3	-3	-1	-2	-3	-3	-3	3	1	-3	1	-1	-3	-2	0	-3	-1	4
30	-2	-2	1	6	-4	0	2	-1	-1	-3	-4	-1	-3	-4	-2	0	-1	-5	-3	-3
31	-3	-3	-4	-5	-3	-2	-3	-3	-3	-3	-2	-3	-2	1	-4	-3	-3	12	2	-3
32	-2	-2	-2	-3	-3	-2	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
33	-1	1	0	0	-3	6	2	-2	0	-3	-2	1	0	-4	-1	0	-1	-2	-2	-2
34	-2	-3	-3	-4	-3	-4	-4	-3	-1	0	0	-3	0	7	-4	-3	-2	1	3	-1
35	0	-3	-3	-3	-1	-2	-3	-3	-3	3	1	-3	1	-1	-3	-2	0	-3	-1	4
36	-1	0	-2	-3	-5	0	0	-4	-1	4	-1	1	-1	-3	5	-2	-3	-2	-4	0
37	-3	1	-2	-3	-4	4	-3	-3	2	-1	-1	0	-1	2	-4	-2	0	-1	5	0
38	-1	2	1	0	-4	1	0	-1	1	-2	-2	2	-2	0	-1	1	-1	0	0	-2
39	-1	-1	1	0	-3	0	0	3	-1	-3	-3	0	-3	-2	1	3	0	0	-2	-3
40	-2	1	1	0	-4	1	-2	3	3	-3	-3	1	-2	-2	-1	1	-2	2	1	-3
41	-1	2	1	0	-3	1	1	1	2	-3	-2	1	-1	-1	-2	1	0	-1	0	-2
42	-2	2	0	-2	-4	2	1	-4	1	-1	-2	2	-1	-2	2	0	1	0	0	-1
43	-1	3	-1	-2	-1	1	-2	-3	0	0	0	1	0	0	-4	-2	0	3	3	1
44	-2	1	2	1	-4	0	-1	-1	0	-3	-2	1	-3	-1	-3	2	2	-1	0	-1
45	-1	1	0	-2	-4	2	1	-1	0	-1	-1	1	-2	-3	0	1	0	-4	-1	2
46	-1	1	0	0	-3	0	0	-2	1	-1	0	0	0	-1	2	0	0	1	2	-1
47	0	0	-2	-1	-1	3	4	1	0	-2	-2	1	-1	-3	-3	0	0	0	-2	-1
48	-2	0	-2	-4	-6	7	-1	-4	-1	-3	-2	-1	1	-4	-2	-2	0	-5	-5	0
49	-2	2	-2	-2	-4	-1	0	-4	1	1	1	1	0	0	-3	-1	2	2	-1	2
50	-1	1	1	1	-4	0	1	0	0	-3	-2	1	-1	-2	3	0	0	-2	-1	-2
51	-2	3	0	-1	-4	2	1	-2	1	-4	-4	2	-3	-4	-1	3	1	-2	-4	-4
52	-1	1	0	-3	-3	0	-3	-4	-1	1	2	0	1	-2	-4	0	1	-4	-2	2
53	-1	-2	-1	-1	-3	-3	-2	-1	2	-3	-1	-2	-1	4	-2	0	0	7	4	-2

Continued on next page

Table 7.8.1 – continued from previous page

S.No.	**A**	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
54	0	-2	-3	-3	-1	0	-1	0	-1	2	1	-2	0	1	0	0	0	-3	2	2
55	-1	1	2	-1	-3	1	0	0	0	-1	-1	1	-2	-1	0	1	1	-3	-1	-1
56	-1	-2	0	2	-1	-1	-1	0	0	-2	0	1	0	2	-2	-1	-1	1	4	0
57	0	-3	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-3	-3	-3
58	-1	-2	-2	-2	-3	-1	-1	-2	-2	-3	-3	-1	-3	-4	8	-1	-1	-4	-3	-3
59	-2	-2	-4	-4	-1	-2	-3	-4	-3	2	4	-3	2	0	-3	-3	-1	-2	-1	1
60	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	2	5	-3	-2	0
61	-3	-3	-4	-5	-3	-2	-3	-3	-3	-3	-2	-3	-2	1	-4	-3	-3	11	2	-3
62	-3	-4	-4	-5	-3	-3	-4	-5	-4	3	4	-4	3	1	-4	-4	-2	-3	-2	2
63	-4	-2	-5	-5	-4	-4	-4	-4	-4	-5	-4	-4	-4	-2	-5	-1	-4	12	-1	-5
64	-2	1	-2	-3	-3	1	-2	-3	1	0	1	-2	1	4	-2	-1	-2	3	4	-1
65	-1	-2	-1	-3	-3	-1	-1	-2	3	-1	-1	-2	0	-3	6	-1	-2	-4	-1	0
66	-2	1	3	-1	-3	3	1	-2	-1	-1	0	2	-1	0	-2	-1	-1	-3	-2	-2
67	-1	3	-1	1	-3	2	0	-3	-1	0	-2	0	-1	-2	-2	0	2	-2	2	0
68	-3	-2	8	1	-4	-2	-2	-2	-1	-5	-5	-2	-4	0	-4	-1	-2	-5	-3	-4
69	-1	1	1	-1	-3	2	2	-2	2	-1	-3	3	-2	-3	-2	0	0	-4	-2	-1
70	-2	-4	-4	-5	-3	-4	-4	-5	-4	6	0	-4	0	-1	-4	-3	-2	-3	2	3
71	-1	0	-1	-1	2	2	2	-2	4	-3	-3	1	-2	-3	-2	1	2	-3	1	-2
72	-1	-1	-1	2	-4	6	2	-3	-1	-3	-3	-1	3	-4	-3	-1	0	-4	-3	-3
73	0	2	-1	-1	-3	0	1	2	1	-3	-3	1	-2	-2	-2	1	1	-2	2	-2
74	-1	1	1	2	-3	1	1	-2	-1	-2	-1	1	1	-2	-2	1	1	-2	1	-2
75	-2	1	3	1	-3	1	1	-1	-1	-1	-1	2	-2	-3	-2	1	-1	-4	-3	-2
76	-1	-1	1	3	-3	1	2	-2	2	-3	-3	2	-3	-3	-2	2	0	-4	-1	-3
77	0	-1	2	1	-3	0	1	2	2	0	-3	2	-2	-3	-2	0	-1	-4	-2	-2
78	1	1	-2	-2	-3	3	0	-1	-2	0	-1	0	1	2	-2	-1	1	4	-1	-1
79	-3	-3	4	6	-4	-2	0	-2	-2	-4	-4	-2	-4	-5	0	0	-2	-5	-4	-1
80	-4	-1	-4	-5	-4	-3	-4	-4	-3	-3	-3	-4	-3	3	-5	-3	0	10	4	-1

Continued on next page

Table 7.8.1 – continued from previous page

S.No.	**A**	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
81	-1	2	1	-1	-3	1	1	-1	4	-2	-2	2	-2	1	-2	0	1	-3	-1	0
82	-1	0	-2	-3	-2	0	-2	-3	-2	1	2	0	2	0	1	-1	0	3	-1	1
83	-2	5	1	-1	-3	0	2	-3	-1	-1	-1	0	-1	-3	-2	-1	0	-3	-2	0
84	-2	-2	0	1	-3	-1	2	-3	3	-2	-1	-2	2	3	2	-1	0	-2	1	-2
85	-1	-1	1	-1	-3	2	2	-2	-1	-2	-2	-1	0	-3	5	1	0	-4	-3	-1
86	-1	-1	-1	-1	-2	0	2	-3	-2	-1	0	1	-1	-3	2	0	3	-3	-2	0
87	0	0	2	2	-3	0	1	0	-1	-1	-2	0	-2	-2	0	0	1	-4	-3	-1
88	-2	0	2	2	-4	-2	0	-2	1	-2	-3	-1	-2	-4	-3	4	3	-5	-4	-3
89	1	-1	0	4	-4	0	2	-2	2	-2	-3	-1	-3	-4	-3	1	-2	-5	-3	1
90	1	2	1	-1	-2	1	0	-1	3	-2	-2	1	0	-3	0	1	0	-3	-1	-1
91	-1	2	1	0	-2	-1	0	-2	-3	-1	-2	0	-2	-3	0	1	2	-2	0	0
92	-2	1	2	1	-3	4	1	-1	3	-3	-2	-1	0	-3	-2	0	0	-4	-2	-2
93	-3	-3	2	7	-5	-2	1	-3	-3	-5	-6	-1	-5	-5	-3	-2	-2	-6	-5	-5
94	-2	-2	-4	-4	-3	-2	-3	-1	-4	0	5	-2	1	-1	-2	-2	-2	-3	-3	1
95	-2	7	-1	-3	-5	2	-1	-4	-1	-5	-4	3	-3	-4	-2	-2	-3	-4	-2	-4
96	-4	-4	-4	-5	-4	-4	-4	-5	-1	-2	0	-4	-2	6	-5	-4	-3	0	7	-3
97	-3	-5	-5	-5	-3	-4	-5	-5	-5	7	0	-4	0	-2	-4	-4	-1	-4	-3	3
98	-3	-4	-5	-5	-3	-4	-5	-5	-4	1	6	-4	1	-1	-5	-4	-3	-3	-3	-1
99	-1	5	-1	-2	-3	0	-1	3	-1	-4	-3	1	-2	-3	-2	-1	-1	-3	-2	-3
100	0	-1	1	5	-2	0	1	-1	-1	-3	-3	-1	-2	-3	-1	2	2	-4	-3	-2
101	-1	-1	0	-2	-2	0	-1	-3	6	-1	1	-1	2	-1	-2	-1	1	-2	1	-1
102	-1	-1	-1	-1	-3	2	1	-2	1	-2	-2	-1	-1	1	-2	1	-1	1	5	-2
103	-2	2	-2	-3	-3	-1	-2	-3	4	-2	-2	-1	-1	2	-3	-2	-2	6	6	-2
104	2	-1	-1	1	-3	1	5	-1	-1	-3	-3	0	-2	-3	-1	0	-1	-3	-2	-2
105	-1	5	2	-1	-3	0	-1	2	-1	-3	-3	1	-2	-3	-2	-1	-1	-3	-2	-3
106	-1	1	2	1	0	0	1	-1	1	-2	-2	2	-2	-1	-1	1	-1	-2	1	-2
107	-2	3	0	-2	1	3	0	-4	0	-2	-1	2	-1	-2	-1	-2	1	-6	-2	1

Continued on next page

Table 7.8.1 – continued from previous page

S.No.	**A**	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
108	0	1	2	0	0	-1	0	-2	2	-2	-2	2	-2	-1	-1	1	-1	-3	2	-3
109	-1	0	3	3	-1	1	1	-2	1	-4	-2	1	-3	-1	-3	1	-1	1	1	-3
110	-2	2	1	1	-2	1	1	-3	1	-1	-2	2	-1	-4	-3	2	2	-4	-2	-1
111	-3	4	-2	-1	-4	0	-2	-3	1	-1	-1	4	-2	1	-4	-1	-2	-3	2	-3
112	1	2	0	-3	2	-2	-1	0	-1	-2	-1	0	1	-1	-3	1	0	4	-1	0
113	0	2	-1	-3	2	2	0	-2	0	-2	-1	3	0	-4	2	-1	-1	-1	-1	-2
114	1	0	-2	-3	-4	-2	-2	1	-4	-2	-1	0	-3	2	-4	-1	0	8	1	-2
115	1	2	0	-2	-1	0	0	0	0	-1	0	-1	1	0	-3	1	0	0	0	-1
116	0	2	-3	-2	0	1	-2	-3	-1	-1	-2	4	-1	-4	-4	1	1	1	-4	0
117	2	5	-2	-3	0	-1	-2	-1	0	-1	-1	-1	-1	0	-3	1	-2	0	0	-2
118	-2	1	6	0	-5	-1	-1	-2	5	-3	-5	-1	-4	-1	-4	-1	0	-5	-2	-2
119	-3	-4	0	-4	-4	-4	-4	-3	-5	-4	-5	-4	-4	-6	-5	0	7	-6	-5	-1
120	-1	1	1	0	-2	0	2	-1	0	-2	-2	2	-2	1	-1	1	-1	-2	2	-1
121	0	0	0	-1	1	0	0	-1	2	-1	0	0	0	1	-2	0	0	1	2	-1
122	-1	-2	0	2	-3	2	-1	-3	-1	3	0	-2	-1	-1	-3	-1	1	-2	2	0
123	-2	6	-2	-2	-4	0	-1	-3	2	-1	-3	2	-2	-3	-3	-2	-2	-3	-2	-3
124	-1	3	-2	-2	0	1	1	-3	-2	1	0	1	-1	0	-3	0	0	2	-2	0
125	-2	-2	-3	-3	-3	-2	-3	-3	1	-1	-1	-2	-1	5	-3	-2	-2	2	6	-1
126	-3	-3	1	0	-3	-4	-2	-5	6	-3	-4	-5	-3	4	-6	-2	-3	-3	7	-3

7.9 ASA features for HB-34

This table includes the solvent accessible surface area (ASA) features for the residues in HB-34 dataset. SBr is relative surface burial, RASA is relative ASA.

Table 7.9.1: ASA features for HB-34 dataset.

S.No.	**ASA(monomer)**	**Δ ASA**	**SBr**	**RASA**	**Δ RASA**
1	51	30	0.59	0.26	0.15
2	39	33	0.85	0.17	0.15
3	42	36	0.86	0.19	0.16
4	58	30	0.52	0.3	0.15
5	3	3	1	0.02	0.02
6	71	54	0.76	0.31	0.24
7	96	52	0.54	0.47	0.25
8	50	7	0.14	0.26	0.04
9	54	32	0.59	0.28	0.16
10	121	117	0.97	0.55	0.53
11	130	98	0.75	0.8	0.6
12	183	176	0.96	0.82	0.79
13	168	36	0.21	0.68	0.15
14	41	32	0.78	0.24	0.19
15	38	36	0.95	0.27	0.25
16	161	70	0.43	0.83	0.36
17	116	102	0.88	0.6	0.53
18	69	58	0.84	0.44	0.37
19	75	53	0.71	0.48	0.34
20	33	26	0.79	0.15	0.12
21	87	67	0.77	0.61	0.47
22	45	38	0.84	0.22	0.19
23	145	45	0.31	0.74	0.23
24	20	12	0.6	0.1	0.06
25	0	0	0	0	0
26	161	0	0	0.79	0
27	75	5	0.07	0.46	0.03
28	35	27	0.77	0.21	0.17
29	78	78	1	0.35	0.35
30	153	112	0.73	0.69	0.5
31	116	94	0.81	0.51	0.41
32	70	54	0.77	0.45	0.34
33	35	35	1	0.22	0.22
34	125	121	0.97	0.56	0.55
35	110	3	0.03	0.54	0.01

Continued on next page

Table 7.9.1 – continued from previous page

S.No.	**ASA(monomer)**	**Δ ASA**	**SBr**	**RASA**	**Δ RASA**
36	109	74	0.68	0.77	0.52
37	155	154	0.99	0.99	0.98
38	123	107	0.87	0.6	0.52
39	28	24	0.86	0.17	0.14
40	74	16	0.22	0.38	0.08
41	32	29	0.91	0.16	0.15
42	157	54	0.34	0.77	0.26
43	113	34	0.3	0.87	0.26
44	29	11	0.38	0.35	0.13
45	152	60	0.39	0.93	0.37
46	97	67	0.69	0.43	0.3
47	87	45	0.52	0.42	0.22
48	45	29	0.64	0.35	0.22
49	36	25	0.69	0.18	0.12
50	160	46	0.29	0.72	0.21
51	53	45	0.85	0.33	0.28
52	18	16	0.89	0.13	0.11
53	107	82	0.77	0.66	0.5
54	17	3	0.18	0.09	0.02
55	80	53	0.66	0.49	0.32
56	63	63	1	0.32	0.32
57	94	30	0.32	0.42	0.14
58	34	31	0.91	0.14	0.13
59	43	1	0.02	0.17	0
60	193	29	0.15	0.97	0.15
61	128	25	0.2	0.66	0.13
62	99	4	0.04	0.44	0.02
63	72	26	0.36	0.55	0.2
64	15	12	0.8	0.11	0.08
65	36	31	0.86	0.28	0.24
66	21	19	0.9	0.12	0.11
67	161	154	0.96	0.71	0.68
68	34	13	0.38	0.2	0.08
69	34	34	1	0.25	0.25
70	103	49	0.48	0.53	0.25
71	71	19	0.27	0.35	0.09
72	12	12	1	0.09	0.09
73	30	23	0.77	0.18	0.14
74	111	75	0.68	0.57	0.39
75	57	28	0.49	0.34	0.17
76	140	36	0.26	0.71	0.18

Continued on next page

Table 7.9.1 – continued from previous page

S.No.	**ASA(monomer)**	**Δ ASA**	**SBr**	**RASA**	**Δ RASA**
77	159	65	0.41	0.78	0.32
78	90	89	0.99	0.4	0.39
79	70	40	0.57	0.49	0.28
80	105	44	0.42	0.42	0.18
81	41	31	0.76	0.32	0.24
82	2	0	0	0.01	0
83	34	13	0.38	0.21	0.08
84	125	105	0.84	0.63	0.53
85	107	95	0.89	0.66	0.58
86	50	50	1	0.35	0.35
87	29	23	0.79	0.18	0.14
88	16	0	0	0.07	0
89	133	122	0.92	0.6	0.55
90	61	49	0.8	0.27	0.22
91	46	3	0.07	0.35	0.02
92	68	27	0.4	0.34	0.14
93	66	39	0.59	0.36	0.21
94	27	27	1	0.13	0.13
95	114	17	0.15	0.73	0.11
96	107	32	0.3	0.54	0.16
97	114	88	0.77	0.56	0.43
98	120	101	0.84	0.61	0.51
99	90	80	0.89	0.69	0.62
100	33	30	0.91	0.2	0.18
101	57	24	0.42	0.4	0.17
102	68	28	0.41	0.43	0.18
103	120	62	0.52	0.48	0.25
104	92	53	0.58	0.71	0.41
105	100	46	0.46	0.61	0.28
106	126	34	0.27	0.64	0.17
107	16	4	0.25	0.08	0.02
108	96	73	0.76	0.52	0.4
109	30	29	0.97	0.16	0.16
110	65	16	0.25	0.33	0.08
111	76	49	0.64	0.39	0.25
112	75	1	0.01	0.46	0.01
113	124	87	0.7	0.56	0.39
114	65	65	1	0.4	0.4
115	161	84	0.52	0.81	0.42
116	13	12	0.92	0.1	0.09
117	67	25	0.37	0.35	0.13

Continued on next page

Table 7.9.1 – continued from previous page

S.No.	**ASA(monomer)**	**Δ ASA**	**SBr**	**RASA**	**Δ RASA**
118	100	77	0.77	0.77	0.59
119	61	39	0.64	0.39	0.25
120	144	32	0.22	0.74	0.16
121	32	29	0.91	0.14	0.13
122	61	39	0.64	0.25	0.16
123	77	71	0.92	0.47	0.44
124	39	31	0.79	0.19	0.15
125	56	49	0.88	0.39	0.35
126	6	6	1	0.03	0.03
127	104	64	0.62	0.42	0.26
128	21	21	1	0.12	0.12
129	103	2	0.02	0.42	0.01
130	199	76	0.38	0.9	0.34
131	19	19	1	0.08	0.08
132	63	26	0.41	0.28	0.11
133	89	12	0.13	0.55	0.07
134	65	0	0	0.4	0
135	77	18	0.23	0.47	0.11
136	116	89	0.77	0.52	0.4
137	22	0	0	0.14	0
138	62	58	0.94	0.33	0.31
139	6	4	0.67	0.07	0.05
140	24	0	0	0.17	0
141	93	50	0.54	0.48	0.26
142	130	79	0.61	0.59	0.36
143	16	0	0	0.19	0
144	71	69	0.97	0.32	0.31
145	116	115	0.99	0.51	0.51
146	66	55	0.83	0.51	0.42
147	103	87	0.84	0.73	0.61
148	167	157	0.94	1.02	0.96
149	26	24	0.92	0.13	0.12
150	161	146	0.91	0.82	0.74
151	90	37	0.41	0.66	0.27
152	90	34	0.38	0.63	0.24
153	152	69	0.45	0.83	0.38
154	72	56	0.78	0.86	0.67
155	122	122	1	0.9	0.9
156	124	36	0.29	0.91	0.26
157	108	108	1	0.56	0.56
158	72	23	0.32	0.36	0.12

Continued on next page

Table 7.9.1 – continued from previous page

S.No.	**ASA(monomer)**	**Δ ASA**	**SBr**	**RASA**	**Δ RASA**
159	136	99	0.73	0.6	0.44
160	128	115	0.9	0.78	0.7
161	83	8	0.1	0.49	0.05
162	80	72	0.9	0.47	0.43
163	92	1	0.01	0.56	0.01
164	86	69	0.8	0.44	0.36
165	85	37	0.44	0.43	0.19
166	130	109	0.84	0.92	0.77
167	71	34	0.48	0.45	0.22
168	167	73	0.44	0.86	0.38
169	25	25	1	0.16	0.16
170	102	101	0.99	0.52	0.51
171	40	15	0.38	0.22	0.08
172	75	75	1	0.34	0.34
173	161	126	0.78	0.65	0.51
174	35	21	0.6	0.15	0.09
175	155	71	0.46	0.63	0.29
176	67	61	0.91	0.41	0.37
177	51	15	0.29	0.36	0.11
178	50	50	1	0.24	0.24
179	100	82	0.82	0.49	0.4
180	7	2	0.29	0.04	0.01
181	40	40	1	0.31	0.31
182	100	99	0.99	0.61	0.61
183	109	72	0.66	0.64	0.43
184	32	19	0.59	0.23	0.13
185	71	19	0.27	0.36	0.1
186	10	8	0.8	0.07	0.06
187	59	15	0.25	0.3	0.08
188	128	61	0.48	0.52	0.25
189	53	10	0.19	0.26	0.05
190	102	96	0.94	0.41	0.39
191	160	133	0.83	0.65	0.54
192	91	89	0.98	0.58	0.57
193	144	144	1	1.01	1.01
194	97	87	0.9	0.75	0.67
195	142	102	0.72	0.57	0.41
196	109	25	0.23	0.59	0.14
197	45	29	0.64	0.23	0.15
198	11	5	0.45	0.06	0.03
199	180	146	0.81	0.73	0.59

Continued on next page

Table 7.9.1 – continued from previous page

S.No.	**ASA(monomer)**	**Δ ASA**	**SBr**	**RASA**	**Δ RASA**
200	192	75	0.39	0.77	0.3
201	105	12	0.11	0.67	0.08
202	101	72	0.71	0.55	0.39
203	181	96	0.53	0.8	0.42
204	84	52	0.62	0.43	0.27
205	85	62	0.73	0.46	0.34
206	62	32	0.52	0.28	0.14
207	80	71	0.89	0.35	0.31
208	119	78	0.66	0.73	0.48
209	97	56	0.58	0.62	0.36
210	72	51	0.71	0.44	0.31
211	28	28	1	0.14	0.14
212	112	69	0.62	0.69	0.42
213	182	155	0.85	0.82	0.7
214	143	67	0.47	0.64	0.3
215	101	5	0.05	0.49	0.02
216	62	31	0.5	0.34	0.17
217	19	0	0	0.14	0
218	85	49	0.58	0.5	0.29
219	208	198	0.95	0.94	0.89
220	90	0	0	0.41	0
221	73	56	0.77	0.36	0.27
222	37	0	0	0.23	0
223	29	2	0.07	0.18	0.01
224	214	170	0.79	0.86	0.69
225	153	77	0.5	0.79	0.4
226	16	10	0.63	0.08	0.05
227	3	3	1	0.01	0.01
228	108	108	1	0.59	0.59
229	112	89	0.79	0.5	0.4
230	117	57	0.49	0.53	0.26
231	108	76	0.7	0.48	0.33
232	47	40	0.85	0.29	0.24
233	62	50	0.81	0.31	0.25
234	84	40	0.48	0.52	0.25
235	109	33	0.3	0.84	0.25
236	76	59	0.78	0.45	0.35
237	31	31	1	0.24	0.24
238	169	48	0.28	0.87	0.25
239	2	1	0.5	0.01	0.01
240	89	20	0.22	0.55	0.12

Continued on next page

Table 7.9.1 – continued from previous page

S.No.	**ASA(monomer)**	**Δ ASA**	**SBr**	**RASA**	**Δ RASA**
241	83	58	0.7	0.43	0.3
242	37	36	0.97	0.18	0.18
243	129	123	0.95	0.58	0.55
244	136	110	0.81	0.66	0.54
245	125	125	1	0.8	0.8
246	45	22	0.49	0.35	0.17
247	172	76	0.44	0.89	0.39
248	84	51	0.61	0.37	0.22
249	41	24	0.59	0.17	0.1
250	174	72	0.41	0.85	0.35
251	163	100	0.61	0.73	0.45
252	135	132	0.98	0.83	0.81
253	92	90	0.98	0.56	0.55
254	71	42	0.59	0.5	0.3
255	40	40	1	0.21	0.21
256	38	16	0.42	0.27	0.11
257	110	5	0.05	0.85	0.04
258	41	8	0.2	0.32	0.06
259	30	29	0.97	0.18	0.18
260	80	58	0.73	0.56	0.41
261	13	13	1	0.09	0.09
262	52	23	0.44	0.37	0.16
263	65	41	0.63	0.34	0.21
264	28	0	0	0.15	0
265	42	7	0.17	0.32	0.05
266	38	38	1	0.29	0.29
267	74	44	0.59	0.45	0.27
268	11	11	1	0.07	0.07
269	98	96	0.98	0.44	0.43
270	176	102	0.58	0.79	0.46
271	229	96	0.42	0.92	0.39
272	18	18	1	0.13	0.13
273	58	45	0.78	0.3	0.23
274	36	36	1	0.28	0.28
275	30	1	0.03	0.22	0.01
276	173	160	0.92	1.05	0.98
277	179	155	0.87	0.91	0.79
278	82	81	0.99	0.58	0.57
279	74	70	0.95	0.38	0.36
280	159	106	0.67	0.72	0.48
281	23	1	0.04	0.09	0

Continued on next page

Table 7.9.1 – continued from previous page

S.No.	**ASA(monomer)**	**Δ ASA**	**SBr**	**RASA**	**Δ RASA**
282	25	7	0.28	0.1	0.03
283	70	42	0.6	0.49	0.3
284	184	180	0.98	0.9	0.88
285	210	199	0.95	0.85	0.8
286	114	69	0.61	0.67	0.41
287	60	26	0.43	0.42	0.18
288	171	70	0.41	0.69	0.28
289	56	31	0.55	0.39	0.22
290	70	69	0.99	0.36	0.36
291	158	152	0.96	0.77	0.74
292	78	75	0.96	0.38	0.37
293	112	82	0.73	0.71	0.52
294	124	42	0.34	0.76	0.26
295	64	56	0.88	0.28	0.25
296	97	73	0.75	0.68	0.51
297	168	167	0.99	1.02	1.02
298	77	56	0.73	0.4	0.29
299	141	122	0.87	0.64	0.55
300	190	100	0.53	0.77	0.4
301	99	55	0.56	0.63	0.35
302	26	26	1	0.15	0.15
303	121	58	0.48	0.59	0.28
304	180	109	0.61	0.81	0.49
305	121	94	0.78	0.74	0.57
306	76	19	0.25	0.47	0.12
307	92	83	0.9	0.54	0.49
308	110	93	0.85	0.56	0.47
309	82	65	0.79	0.5	0.4
310	147	0	0	0.72	0
311	137	64	0.47	0.69	0.32
312	81	81	1	0.57	0.57
313	192	138	0.72	0.85	0.61

7.10 ASA features for BID-18

This table includes the solvent accessible surface area (ASA) features for the residues in BID-18 dataset. SBr is relative surface burial, RASA is relative ASA.

Table 7.10.1: ASA features for BID-18 dataset.

S.No.	**ASA(monomer)**	**Δ ASA**	**SBr**	**RASA**	**Δ RASA**
1	52	24	0.46	0.26	0.12
2	50	49	0.98	0.25	0.25
3	43	42	0.98	0.22	0.21
4	163	96	0.59	0.8	0.47
5	190	188	0.99	0.84	0.83
6	146	107	0.73	0.71	0.52
7	29	29	1	0.35	0.35
8	125	90	0.72	0.5	0.36
9	109	100	0.92	0.64	0.59
10	40	6	0.15	0.48	0.07
11	172	144	0.84	0.69	0.58
12	148	138	0.93	0.9	0.84
13	66	66	1	0.79	0.79
14	37	27	0.73	0.22	0.16
15	24	24	1	0.17	0.17
16	27	12	0.44	0.14	0.06
17	58	57	0.98	0.35	0.35
18	80	30	0.38	0.62	0.23
19	119	9	0.08	0.61	0.05
20	145	64	0.44	0.79	0.35
21	38	21	0.55	0.2	0.11
22	84	50	0.6	0.65	0.38
23	30	22	0.73	0.18	0.13
24	53	42	0.79	0.32	0.26
25	80	30	0.38	0.75	0.28
26	113	96	0.85	0.69	0.59
27	43	13	0.3	0.26	0.08
28	221	146	0.66	0.89	0.59
29	32	21	0.66	0.23	0.15
30	98	80	0.82	0.6	0.49
31	136	90	0.66	0.6	0.4
32	100	100	1	0.45	0.45
33	126	11	0.09	0.64	0.06
34	118	91	0.77	0.6	0.46
35	70	18	0.26	0.49	0.13
36	108	25	0.23	0.64	0.15
37	24	13	0.54	0.12	0.07
38	97	25	0.26	0.75	0.19
39	73	54	0.74	0.54	0.4
40	132	123	0.93	0.67	0.62

Continued on next page

Table 7.10.1 – continued from previous page

S.No.	**ASA(monomer)**	**Δ ASA**	**SBr**	**RASA**	**Δ RASA**
41	84	10	0.12	0.43	0.05
42	36	20	0.56	0.22	0.12
43	18	17	0.94	0.07	0.07
44	96	49	0.51	0.68	0.35
45	134	34	0.25	0.54	0.14
46	104	95	0.91	0.47	0.43
47	5	3	0.6	0.03	0.02
48	87	45	0.52	0.42	0.22
49	73	69	0.95	0.43	0.41
50	110	54	0.49	0.59	0.29
51	152	60	0.39	0.74	0.29
52	33	25	0.76	0.16	0.12
53	141	90	0.64	0.72	0.46
54	95	95	1	0.51	0.51
55	62	36	0.58	0.44	0.25
56	14	12	0.86	0.07	0.06
57	21	16	0.76	0.25	0.19
58	139	40	0.29	1.02e+14	0.29
59	165	74	0.45	1.01e+14	0.45
60	99	78	0.79	0.7	0.55
61	150	132	0.88	0.66	0.58
62	78	65	0.83	0.55	0.46
63	88	78	0.89	0.39	0.34
64	122	112	0.92	0.62	0.57
65	89	89	1	0.65	0.65
66	91	3	0.03	0.44	0.01
67	73	42	0.58	0.51	0.3
68	13	11	0.85	0.08	0.07
69	87	75	0.86	0.42	0.37
70	40	33	0.83	0.24	0.2
71	72	19	0.26	0.37	0.1
72	31	29	0.94	0.16	0.15
73	130	21	0.16	0.63	0.1
74	116	43	0.37	0.89	0.33
75	148	62	0.42	0.91	0.38
76	39	19	0.49	0.3	0.15
77	46	29	0.63	0.22	0.14
78	135	96	0.71	0.69	0.49
79	59	51	0.86	0.36	0.31
80	85	82	0.96	0.38	0.37
81	79	7	0.09	0.41	0.04

Continued on next page

Table 7.10.1 – continued from previous page

S.No.	**ASA(monomer)**	**Δ ASA**	**SBr**	**RASA**	**Δ RASA**
82	51	44	0.86	0.31	0.27
83	138	69	0.5	0.56	0.28
84	72	32	0.44	0.37	0.16
85	81	59	0.73	0.57	0.42
86	66	37	0.56	0.46	0.26
87	202	128	0.63	0.81	0.52
88	125	82	0.66	0.64	0.42
89	66	66	1	0.36	0.36
90	156	92	0.59	0.85	0.5
91	74	60	0.81	0.57	0.46
92	12	12	1	0.14	0.14
93	92	70	0.76	0.56	0.43
94	82	46	0.56	0.5	0.28
95	213	138	0.65	0.86	0.56
96	162	125	0.77	0.82	0.63
97	106	100	0.94	0.63	0.59
98	220	206	0.94	1.34e+14	1.25e+14
99	125	78	0.62	0.5	0.31
100	83	59	0.71	0.51	0.36
101	79	77	0.97	0.43	0.42
102	101	98	0.97	0.45	0.44
103	120	102	0.85	0.54	0.46
104	137	35	0.26	0.71	0.18
105	111	93	0.84	0.45	0.38
106	109	13	0.12	0.69	0.08
107	156	112	0.72	0.63	0.45
108	139	120	0.86	0.76	0.65
109	180	39	0.22	0.88	0.19
110	38	33	0.87	0.2	0.17
111	130	126	0.97	0.66	0.64
112	83	30	0.36	0.4	0.15
113	23	16	0.7	0.11	0.08
114	96	52	0.54	0.42	0.23
115	87	76	0.87	0.35	0.31
116	13	10	0.77	0.06	0.05
117	95	23	0.24	0.38	0.09
118	46	34	0.74	0.25	0.18
119	4	4	1	0.03	0.03
120	148	137	0.93	0.67	0.62
121	99	87	0.88	0.73	0.64
122	123	120	0.98	0.75	0.74

Continued on next page

Table 7.10.1 – continued from previous page

S.No.	ASA(monomer)	Δ ASA	SBr	RASA	Δ RASA
123	80	74	0.93	0.32	0.3
124	106	41	0.39	0.43	0.17
125	128	79	0.62	0.65	0.4
126	66	41	0.62	0.3	0.18

7.11 Solvent Exposure features for HB-34

This table includes the seven solvent exposure based features calculated for the residues in HB-34 dataset. Here, HSEBD is the number of C_β atoms in the lower half sphere, HSEAU is number of C_α atoms in the upper sphere, HSEAD is the number of C_α atoms in the lower sphere, HSEBU is the number of C_β atoms in the upper sphere, CN is the coordination number, RD is the residue depth and RDa is the C_α atom depth.

Table 7.11.1: Solvent exposure features for HB-34.

S.No.	HSEBD	HSEAU	HSEAD	HSEBU	CN	RD	RDa
1	22	21	24	23	45	2.9	3.74
2	20	23	23	26	46	2.94	4.56
3	23	22	27	26	49	3.52	4.28
4	14	10	29	25	39	3.02	3.59
5	22	25	23	26	48	6.18	6.45
6	23	24	25	26	49	3.78	5.46
7	24	23	24	23	47	2.61	3.49
8	13	16	24	27	40	2.5	2.73
9	14	17	23	26	40	2.69	2.91
10	21	21	24	24	45	3.46	4.35
11	23	21	24	22	45	2.06	2.72
12	23	24	28	29	52	2.48	2
13	10	5	27	22	32	2.1	2.31
14	19	19	20	20	39	2.33	2
15	13	12	36	35	48	5.66	5.41
16	12	14	23	25	37	1.76	2
17	31	27	29	25	56	2.66	3.22
18	29	28	28	27	56	2.17	2
19	20	17	17	14	34	1.82	2
20	26	26	29	29	55	3.24	4.83
21	27	26	26	25	52	2.23	2.82
22	23	24	18	19	42	3.83	4.41
23	17	24	18	25	42	1.93	2.37

Continued on next page

Table 7.11.1 – continued from previous page

S.No.	HSEBD	HSEAU	HSEAD	HSEBU	CN	RD	RDa
24	29	27	22	20	49	2.31	2.37
25	31	29	25	23	54	5.21	5.44
26	7	0	32	25	32	2.15	2
27	23	27	9	13	36	2.27	2
28	24	22	26	24	48	2.76	2.86
29	25	22	31	28	53	6.46	6
30	20	17	23	20	40	2.62	3.18
31	25	24	24	23	48	6.7	6.77
32	20	21	26	27	47	2.93	2.58
33	26	24	31	29	55	5.88	5.85
34	25	23	19	17	42	6.71	6.43
35	15	11	27	23	38	2.78	2.74
36	18	21	30	33	51	2.95	3.38
37	26	28	27	29	55	5.71	6.08
38	23	24	30	31	54	5.08	5.11
39	16	16	34	34	50	5.19	5.84
40	7	13	28	34	41	2.41	2.71
41	24	22	29	27	51	6.65	7.01
42	7	11	20	24	31	1.99	2.47
43	4	8	17	21	25	1.85	2.31
44	16	11	18	13	29	2.4	2
45	17	19	16	18	35	1.77	2.36
46	24	21	20	17	41	4.28	2.66
47	16	15	26	25	41	2.45	2
48	22	24	18	20	42	4.27	4.85
49	24	20	26	22	46	3.02	2.93
50	7	12	18	23	30	1.96	2.69
51	19	22	27	30	49	5.66	5.28
52	28	28	23	23	51	5.06	5.22
53	21	20	26	25	46	1.99	2
54	26	16	29	19	45	3.03	2.99
55	10	8	28	26	36	3.36	3.94
56	20	14	35	29	49	5.54	6.42
57	11	11	18	18	29	2.57	2.86
58	27	30	24	27	54	7.27	5.84
59	16	18	23	25	41	2.95	4.27
60	1	12	13	24	25	1.79	2
61	5	8	17	20	25	1.91	2.47
62	13	15	10	12	25	1.98	2.33
63	7	15	10	18	25	1.89	2.48
64	24	27	19	22	46	4.63	4.1

Continued on next page

Table 7.11.1 – continued from previous page

S.No.	**HSEBD**	**HSEAU**	**HSEAD**	**HSEBU**	**CN**	**RD**	**RDa**
65	15	16	26	27	42	3.83	3.87
66	29	31	18	20	49	4.19	3.78
67	22	22	25	25	47	3.11	2.7
68	22	21	26	25	47	2.99	2.69
69	24	21	24	21	45	5.24	5.88
70	14	19	19	24	38	1.86	2.31
71	23	18	24	19	42	2.6	2.55
72	21	21	28	28	49	4.83	5.21
73	26	29	21	24	50	5.84	6.42
74	17	13	32	28	45	2.08	2.62
75	21	24	18	21	42	2.5	2.81
76	7	10	22	25	32	1.79	2
77	11	9	26	24	35	2.07	2.11
78	23	23	25	25	48	5.88	5.82
79	20	22	27	29	49	3.37	3.74
80	26	19	31	24	50	2.39	2.2
81	30	28	21	19	49	2.69	2.34
82	29	26	24	21	50	7.32	6.53
83	16	18	17	19	35	2.84	2.97
84	25	21	13	9	34	3.45	2
85	17	16	23	22	39	3.96	3.16
86	25	27	26	28	53	7.49	8.42
87	24	25	19	20	44	5.12	4.69
88	33	33	15	15	48	4.88	4.13
89	29	23	24	18	47	5.05	3.91
90	32	27	12	7	39	4	2.79
91	22	26	15	19	41	2.25	2.67
92	14	17	30	33	47	2.37	2.6
93	24	25	31	32	56	4.33	6.15
94	21	22	29	30	51	4.32	5.27
95	12	6	19	13	25	1.84	2
96	12	16	19	23	35	2.46	2.61
97	27	28	21	22	49	6.04	5.98
98	32	28	25	21	53	4.97	5.33
99	16	22	23	29	45	4.34	5.11
100	24	22	30	28	52	3.17	3.47
101	33	33	26	26	59	5.72	4.96
102	12	19	15	22	34	1.92	2.58
103	20	26	21	27	47	2.23	2.71
104	11	14	22	25	36	2.42	2
105	13	11	24	22	35	1.97	2

Continued on next page

Table 7.11.1 – continued from previous page

S.No.	HSEBD	HSEAU	HSEAD	HSEBU	CN	RD	RDa
106	14	5	23	14	28	2.19	2
107	17	13	28	24	41	2.32	2.26
108	14	18	24	28	42	2.42	3.19
109	25	27	22	24	49	5.68	6.36
110	13	8	31	26	39	2.53	3.02
111	18	17	27	26	44	2.98	3.96
112	15	9	31	25	40	1.86	2
113	11	15	19	23	34	2.35	2.8
114	23	22	18	17	40	5.12	4.91
115	15	16	20	21	36	1.95	2.44
116	22	27	14	19	41	3.44	2.94
117	23	27	21	25	48	2.03	2.36
118	18	21	21	24	42	2.37	2
119	12	18	20	26	38	3.36	4.53
120	7	3	24	20	27	1.83	2
121	25	24	29	28	53	5.81	6.3
122	25	22	29	26	51	5.39	7.95
123	28	28	28	28	56	8.25	8.62
124	31	23	36	28	59	4.86	6.11
125	27	27	33	33	60	7.1	6.59
126	29	24	30	25	54	6.44	7.36
127	23	27	24	28	51	3.17	4.21
128	25	19	30	24	49	3.16	3.15
129	17	10	24	17	34	1.83	2
130	6	4	22	20	26	1.82	2
131	24	28	26	30	54	6.67	6.33
132	22	20	31	29	51	6	6.62
133	14	16	24	26	40	5.12	5.96
134	13	19	14	20	33	2.58	2.99
135	6	3	23	20	26	1.86	2
136	15	10	20	15	30	2.07	2
137	24	15	26	17	41	2.17	2
138	29	28	20	19	48	3.32	5.12
139	5	10	17	22	27	1.81	2
140	27	24	25	22	49	5.25	5.25
141	13	8	24	19	32	2.31	2
142	15	9	31	25	40	2.18	2
143	10	14	28	32	42	1.85	2
144	30	29	28	27	57	4.65	5.8
145	24	26	25	27	51	4.32	5.69
146	19	17	30	28	47	2.31	2.87

Continued on next page

Table 7.11.1 – continued from previous page

S.No.	**HSEBD**	**HSEAU**	**HSEAD**	**HSEBU**	**CN**	**RD**	**RDa**
147	23	27	24	28	51	2.83	3.46
148	33	33	24	24	57	4.99	4.95
149	27	24	27	24	51	5.3	4.89
150	23	27	20	24	47	3.96	2.68
151	7	8	17	18	25	1.98	2
152	16	16	15	15	31	1.84	2.16
153	17	17	25	25	42	1.96	2.75
154	30	31	22	23	53	1.61	2
155	34	30	23	19	53	2.95	2.72
156	10	8	16	14	24	1.89	2
157	22	22	25	25	47	2.52	2
158	9	13	12	16	25	2.06	2.62
159	3	0	16	13	16	1.67	2
160	21	14	21	14	35	2.43	2
161	15	7	21	13	28	1.96	2
162	23	22	25	24	47	3.24	3.19
163	10	5	27	22	32	2.34	2.18
164	19	17	24	22	41	3.62	5.41
165	19	19	12	12	31	1.9	2.23
166	12	22	20	30	42	4.26	4.32
167	13	12	24	23	36	2.3	3.09
168	7	10	31	34	41	2.18	2.88
169	24	27	23	26	50	5.25	5.55
170	32	30	30	28	60	7.36	7.5
171	20	16	28	24	44	2.63	2.72
172	28	30	15	17	45	3.67	2
173	19	19	18	18	37	2.01	2
174	25	25	22	22	47	3.56	5.17
175	10	9	22	21	31	2.01	2
176	25	21	22	18	43	2.21	2
177	10	15	24	29	39	2.33	2.72
178	27	27	27	27	54	3.98	3.79
179	26	23	27	24	50	2.75	2.7
180	23	23	22	22	45	3.44	4.09
181	30	27	22	19	49	2.74	2
182	21	19	18	16	37	3.09	3.28
183	19	19	19	19	38	2.72	2.64
184	20	22	20	22	42	3.16	3.51
185	16	14	27	25	41	1.97	2
186	25	23	19	17	42	5.48	5.24
187	19	12	26	19	38	2.74	3.72

Continued on next page

Table 7.11.1 – continued from previous page

S.No.	HSEBD	HSEAU	HSEAD	HSEBU	CN	RD	RDa
188	14	8	29	23	37	3.01	3.81
189	20	17	28	25	45	3.53	4.26
190	28	21	27	20	48	6.11	6.25
191	20	15	32	27	47	4.82	7.32
192	22	20	30	28	50	3.24	3.91
193	29	29	27	27	56	30.73	30.7
194	24	35	19	30	54	27.56	28.1
195	21	19	16	14	35	2.14	2
196	17	10	21	14	31	2.23	2
197	23	21	23	21	44	4.61	5.28
198	29	28	24	23	52	5.98	5.5
199	21	18	21	18	39	2.03	2
200	19	14	22	17	36	2.01	2.29
201	15	14	17	16	31	1.89	2
202	17	17	26	26	43	2.16	2
203	10	12	20	22	32	2	2.8
204	18	17	17	16	34	1.79	2
205	22	18	32	28	50	3.66	3.83
206	25	20	27	22	47	3.49	4.44
207	18	21	22	25	43	5.67	5.68
208	19	16	22	19	38	2.36	2
209	14	11	25	22	36	2.29	2
210	19	12	27	20	39	2.43	2.29
211	24	27	22	25	49	8.63	7.91
212	19	21	28	30	49	6.37	6.88
213	25	22	28	25	50	8.72	8.77
214	22	22	25	25	47	4.71	5.11
215	5	12	22	29	34	2.27	2.79
216	22	20	33	31	53	7.09	7
217	11	13	25	27	38	3.75	4.62
218	25	24	31	30	55	6.28	5.56
219	29	29	23	23	52	5.72	6.6
220	22	20	21	19	41	5.88	4.81
221	22	25	17	20	42	3.97	3.57
222	21	14	21	14	35	3.83	3.73
223	19	24	14	19	38	3.53	4.28
224	20	13	19	12	32	3.05	2.54
225	15	15	11	11	26	2.38	2
226	23	27	24	28	51	8.16	8.94
227	28	28	22	22	50	8.51	8.92
228	19	25	17	23	42	5.41	5.2

Continued on next page

Table 7.11.1 – continued from previous page

S.No.	HSEBD	HSEAU	HSEAD	HSEBU	CN	RD	RDa
229	17	23	24	30	47	4.21	6.3
230	11	15	27	31	42	2.56	3.37
231	26	23	21	18	44	2.89	3.76
232	17	17	30	30	47	3.54	4.33
233	23	22	24	23	46	4.91	6.97
234	4	14	21	31	35	2.86	3.2
235	7	7	27	27	34	2.27	2
236	21	21	22	22	43	4.79	4.76
237	27	25	21	19	46	6.33	6.64
238	10	8	21	19	29	1.83	2
239	16	15	37	36	52	4.84	4.83
240	11	14	24	27	38	2.1	2.69
241	14	25	20	31	45	2.55	2.54
242	20	15	27	22	42	3.53	2
243	21	22	18	19	40	2.74	2.65
244	22	19	28	25	47	3.09	3.8
245	28	23	27	22	50	4.42	4.3
246	19	20	25	26	45	2.4	2
247	12	15	20	23	35	1.77	2.31
248	19	15	31	27	46	3.46	5.73
249	14	12	34	32	46	3.7	5.24
250	9	12	24	27	36	1.91	2.33
251	17	12	18	13	30	2.44	2
252	24	21	24	21	45	7.15	7.48
253	32	27	28	23	55	7.83	7.32
254	25	23	25	23	48	2.97	2.41
255	21	23	19	21	42	5.11	5.09
256	22	16	11	5	27	2.29	2
257	0	0	20	20	20	1.9	2
258	4	3	23	22	26	2.34	2
259	28	25	16	13	41	4.84	3.88
260	18	22	23	27	45	2.5	3.01
261	33	31	21	19	52	5.65	5.68
262	13	13	26	26	39	2.84	3.45
263	16	17	21	22	38	2.26	2.67
264	28	25	11	8	36	2.44	2.74
265	13	14	16	17	30	2.24	2
266	25	29	19	23	48	5.74	5.77
267	13	16	26	29	42	3.64	4.71
268	30	35	21	26	56	6.88	6.74
269	28	28	22	22	50	5.35	3.77

Continued on next page

Table 7.11.1 – continued from previous page

S.No.	HSEBD	HSEAU	HSEAD	HSEBU	CN	RD	RDa
270	16	13	23	20	36	2.37	2
271	10	16	16	22	32	2.85	4.69
272	30	29	22	21	51	5.66	5.48
273	16	16	20	20	36	2.58	2.58
274	26	29	20	23	49	5.7	5.78
275	23	23	13	13	36	2.22	2.76
276	27	28	20	21	48	7.12	6.91
277	26	25	20	19	45	5.04	4.5
278	30	31	33	34	64	6.77	7.42
279	25	29	33	37	62	6.38	7.83
280	24	25	21	22	46	4.78	7.24
281	17	17	18	18	35	4.82	5.42
282	24	22	27	25	49	6.44	6.97
283	17	21	18	22	39	2.87	2.5
284	39	33	31	25	64	6.94	6.62
285	32	26	33	27	59	3.12	3.95
286	22	22	29	29	51	2.4	2.7
287	24	25	24	25	49	2.38	3.17
288	15	8	25	18	33	2.23	2
289	14	13	27	26	40	2.23	2.54
290	21	28	24	31	52	5.16	5.71
291	23	24	27	28	51	2.37	2.64
292	26	32	21	27	53	4.13	4.86
293	14	14	16	16	30	1.92	2
294	8	12	22	26	34	1.87	2.72
295	27	30	13	16	43	3.31	2.33
296	29	27	31	29	58	3.27	2.88
297	39	34	28	23	62	6.4	6.6
298	23	26	32	35	58	3.97	5.15
299	30	28	30	28	58	3.48	4.77
300	17	12	32	27	44	2.29	2
301	8	10	25	27	35	2.1	2.65
302	23	20	22	19	42	4.96	5.07
303	14	18	19	23	37	2.16	2
304	13	14	14	15	28	2.54	2.67
305	28	23	13	8	36	3.62	3.17
306	5	3	28	26	31	2.26	2
307	20	29	10	19	39	4.31	4.5
308	24	20	12	8	32	2.45	2.72
309	16	18	14	16	32	2.77	3.09
310	12	5	12	5	17	1.89	2

Continued on next page

Table 7.11.1 – continued from previous page

S.No.	**HSEBD**	**HSEAU**	**HSEAD**	**HSEBU**	**CN**	**RD**	**RDa**
311	18	15	7	4	22	1.92	2
312	24	23	13	12	36	4.63	5.15
313	18	20	12	14	32	2.21	2

7.12 Solvent Exposure features for BID-18

This table includes the seven solvent exposure based features calculated for the residues in BID-18 dataset. Here, HSEBD is the number of C_β atoms in the lower half sphere, HSEAU is number of C_α atoms in the upper sphere, HSEAD is the number of C_α atoms in the lower sphere, HSEBU is the number of C_β atoms in the upper sphere, CN is the coordination number, RD is the residue depth and RDa is the C_α atom depth.

Table 7.12.1: Solvent exposure features for BID-18.

S.No.	**HSEBD**	**HSEAU**	**HSEAD**	**HSEBU**	**CN**	**RD**	**RDa**
1	16	18	21	23	39	3.33	4.53
2	30	28	20	18	48	5.47	3.97
3	29	32	18	21	50	4.37	3.13
4	20	19	22	21	41	2.38	2
5	22	30	19	27	49	4.99	5.43
6	20	17	20	17	37	2.88	2.79
7	19	23	26	30	49	5.63	6
8	26	20	29	23	49	2.51	2
9	26	26	25	25	51	2.47	2.85
10	20	15	24	19	39	2.31	2
11	23	20	20	17	40	2.05	2
12	30	24	16	10	40	2.45	2
13	15	21	16	22	37	3.21	3.18
14	12	19	11	18	30	2.09	2.32
15	15	11	23	19	34	1.86	2.28
16	32	32	23	23	55	4.51	4.67
17	17	20	27	30	47	4.95	4.63
18	30	27	27	24	54	4.99	4.96
19	18	24	21	27	45	2.62	3.12
20	12	12	30	30	42	2.29	2.81
21	8	12	12	16	24	1.78	2.27
22	5	6	23	24	29	1.71	2
23	24	21	11	8	32	1.92	2
24	30	26	31	27	57	3.96	2

Continued on next page

Table 7.12.1 – continued from previous page

S.No.	HSEBD	HSEAU	HSEAD	HSEBU	CN	RD	RDa
25	13	0	0	18	31	1.83	2
26	26	19	17	10	36	2.28	2
27	20	17	9	6	26	1.86	2
28	11	10	17	16	27	1.82	2
29	19	23	11	15	34	2.14	2.33
30	20	14	16	10	30	2.35	2
31	17	11	23	17	34	2.53	2
32	30	31	11	12	42	4.88	3.9
33	10	1	27	18	28	1.79	2
34	25	18	13	6	31	2.39	2
35	12	15	9	12	24	1.91	2.25
36	6	10	27	31	37	1.92	2
37	26	28	25	27	53	3.81	4.95
38	16	20	18	22	38	2.01	2.54
39	11	8	26	23	34	1.83	2.29
40	12	8	26	22	34	2.78	2.92
41	27	27	17	17	44	2.82	3.59
42	11	3	13	5	16	1.86	2
43	12	12	6	6	18	1.95	2
44	25	24	17	16	41	3.66	3.51
45	24	20	30	26	50	3.24	3.03
46	34	36	23	25	59	5.24	5.39
47	29	27	25	23	52	3.11	2.39
48	12	20	22	30	42	2.22	2
49	10	9	30	29	39	2.82	2.63
50	11	18	19	26	37	1.94	2
51	21	17	17	13	34	2.37	2
52	31	30	23	22	53	6.08	6.75
53	16	14	24	22	38	2.17	2
54	21	16	23	18	39	4.6	4.94
55	14	21	19	26	40	2.15	2.7
56	30	32	11	13	43	3.08	2.09
57	11	11	14	14	25	2.1	2
58	4	9	16	21	25	1.96	2
59	7	16	15	24	31	2.25	2.42
60	14	19	12	17	31	1.91	2.53
61	22	22	15	15	37	2.52	2.64
62	26	23	12	9	35	3.17	2.95
63	16	16	17	17	33	2.32	2
64	30	30	28	28	58	3.16	2.34
65	36	33	30	27	63	4.66	4.73

Continued on next page

Table 7.12.1 – continued from previous page

S.No.	HSEBD	HSEAU	HSEAD	HSEBU	CN	RD	RDa
66	19	14	28	23	42	2.16	2
67	20	22	29	31	51	2.79	3.06
68	28	30	29	31	59	6.17	6.27
69	22	24	30	32	54	5.53	5.74
70	19	17	34	32	51	4.52	4.82
71	7	11	28	32	39	2.64	3.02
72	21	22	28	29	50	4.52	5.64
73	4	7	22	25	29	1.91	2
74	5	9	16	20	25	1.96	2.49
75	17	19	18	20	37	1.87	2.4
76	22	23	20	21	43	3.86	4.55
77	27	23	26	22	49	3.03	2.74
78	16	16	25	25	41	2.43	2.32
79	18	21	28	31	49	6.49	6.99
80	19	21	15	17	36	3.28	3.67
81	12	9	24	21	33	2.5	2.77
82	35	27	28	20	55	5.21	5.56
83	13	12	17	16	29	2.16	2.71
84	18	19	12	13	31	2.36	2.52
85	21	20	23	22	43	2.34	2
86	16	21	15	20	36	1.87	2.24
87	9	16	22	29	38	1.98	2.23
88	26	22	29	25	51	2.22	2
89	28	28	25	25	53	5.6	5.11
90	15	0	0	17	32	1.97	2.44
91	18	24	14	20	38	2.19	2.37
92	24	22	25	23	47	5.61	5.24
93	16	16	9	9	25	2.04	2.56
94	8	20	11	23	31	2.33	2.7
95	18	21	23	26	44	2.52	2.98
96	22	27	19	24	46	3.31	4.71
97	19	21	26	28	47	2.6	3.18
98	30	0	0	18	48	2.27	2
99	22	22	28	28	50	2.55	2.98
100	19	13	28	22	41	3.42	4.22
101	20	20	31	31	51	6.69	6.26
102	23	22	36	35	58	4.32	5.31
103	26	26	35	35	61	6.96	8.48
104	21	15	14	8	29	1.78	2
105	23	23	29	29	52	5.09	7.09
106	7	8	19	20	27	1.86	2

Continued on next page

Table 7.12.1 – continued from previous page

S.No.	HSEBD	HSEAU	HSEAD	HSEBU	CN	RD	RDa
107	15	19	23	27	42	3.74	5.47
108	13	20	19	26	39	3.04	3.42
109	11	7	27	23	34	2.1	2
110	21	20	26	25	46	3.62	4.29
111	20	22	22	24	44	3.55	3.84
112	20	20	29	29	49	152.78	150.77
113	17	25	16	24	41	146.64	145.12
114	23	17	28	22	45	154.71	152.41
115	21	21	29	29	50	153.52	150.83
116	21	25	21	25	46	144.65	143.68
117	24	17	34	27	51	155.84	154.52
118	29	19	33	23	52	151.34	151.12
119	26	27	27	28	54	5.93	6.27
120	17	22	18	23	40	2.14	2
121	23	24	23	24	47	2.77	3.01
122	25	39	16	30	55	2.73	2.41
123	28	25	21	18	46	4.37	3.16
124	15	18	27	30	45	2.4	3.02
125	3	2	8	7	10	2.16	2
126	20	21	27	28	48	3.38	4.63

7.13 FASTA Sequence information of residues in HB-34

FASTA format [LP85] is a text-based format for representing either nucleotide sequences or peptide sequences, where base pairs or amino acids are represented using single-letter codes. A sequence in FASTA format begins with a single-line description that is followed by lines of sequence data. The description line is distinguished from the sequence data by a greater-than (">") symbol as the first symbol in the line.

>1DFJ|Chain I|RIBONUCLEASE INHIBITOR|Sus scrofa(9823)
XMNLDIHCEQLSDARWTELLPLLQQYEVVRLDDCGLTEEHCKDIGSALRANP
SLTELCLRTNELGDAGVHLVLQGLQSPTCKIQKLSLQNCSLTEAGCGVLPSTL
RSLPTLRELHLSDNPLGDAGLRLLCEGLLDPQCHLEKLQLEYCRLTAASCEPL
ASVLRATRALKELTVSNNDIGEAGARVLGQGLADSACQLETLRLENCGLTPA
NCKDLCGIVASQASLRELDLGSNGLGDAGIAELCPGLLSPASRLKTLWLWEC
DITASGCRDLCRVLQAKETLKELSLAGNKLGDEGARLLCESLLQPGCQLESL
WVKSCSLTAACCQHVSLMLTQNKHLLELQLSSNKLGDSGIQELCQALSQPGT
TLRVLCLGDCEVTNSGCSSLASLLLANRSLRELDLSNNCVGDPGVLQLLGSLE
QPGCALEQLVLYDTYWTEEVEDRLQALEGSKPGLRVIS

>2PCC|Chains A,C|CYTOCHROME C PEROXIDASE|Saccharomyces cerevisiae(4932)
MITTPLVHVASVEKGRSYEDFQKVYNAIALKLREDDEYDNYIGYGPVLVRLA
WHISGTWDKHDNTGGSYGGTYRFKKEFNDPSNAGLQNGFKFLEPIHKEFPW
ISSGDLFSLGGVTAVQEMQGPKIPWRCGRVDTPEDTTPDNGRLPDADKDAG
YVRTFFQRLNMNDREVVALMGAHALGKTHLKNSGYEGPWGAANNVFTNEF
YLNLLNEDWKLEKNDANNEQWDSKSGYMMLPTDYSLIQDPKYLSIVKEYAN
DQDKFFKDFSKAFEKLLENGITFPKDAPSPFIFKTLEEQGL

>1JTG|Chains A,C|BETA-LACTAMASE TEM|Escherichia coli(562)
HPETLVKVKDAEDQLGARVGYIELDLNSGKILESFRPEERFPMMSTFKVLLC
GAVLSRIDAGQEQLGRRIHYSQNDLVEYSPVTEKHLTDGMTVRELCSAAITM
SDNTAANLLLTTIGGPKELTAFLHNMGDHVTRLDRWEPELNEAIPNDERDTT
MPVAMATTLRKLLTGELLTLASRQQLIDWMEADKVAGPLLRSALPAGWFIA
DKSGAGERGSRGIIAALGPDGKPSRIVVIYTTGSQATMDERNRQIAEIGASLIK
HW

>1JCK|Chains B,D|STAPHYLOCOCCAL ENTEROTOXIN C3|
>Staphylococcus aureus(1280)
ESQPDPMPDDLHKSSEFTGTMGNMKYLYDDHYVSATKVKSVDKFLAHDLIY
NINDKKLNNYDKVKTELLNEDLANKYKDEVVDVYGSNYYVNCYFSSKDNVG
KVTSGKTCMYGGITKHEGNHFDNGNLQNVLIRVYENKRNTISFEVQTDKKS
VTAQELDIKARNFLINKKNLYEFNSSPYETGYIKFIESNGNTFWYDMMPAPG
DKFDQSKYLMIYKDNKMVDSKSVKIEVHLTTKNG

>1JCK|Chains A,C|14.3.D T CELL ANTIGEN RECEPTOR|
>Mus musculus(10090)
AVTQSPRNKVAVTGGKVTLSCQQTNNHNNMYWYRQDTGHGLRLIHYSYGA
GSTEKGDIPDGYKASRPSQEQFSLILELATPSQTSVYFCASGGGRGSYAEQFF
GPGTRLTVLEDLRQVTPPKVSLFEPSKAEIANKQKATLVCLARGFFPDHVEL
SWWVNGKEVHSGVSTDPQAYKESNYSYCLSSRLRVSATFWHNPRNHFRCQV
QFHGLSEEDKWPEGSPKPVTQNISAEAWGRAD

>1DAN|Chain H|BLOOD COAGULATION FACTOR VIIA heavy chain|Homo sapiens(9606)
IVGGKVCPKGECPWQVLLLVNGAQLCGGTLINTIWVVSAAHCFDKIKNWRN
LIAVLGEHDLSEHDGDEQSRRVAQVIIPSTYVPGTTNHDIALLRLHQPVVLTD
HVVPLCLPERTFSERTLAFVRFSLVSGWGQLLDRGATALELMVLNVPRLMT
QDCLQQSRKVGDSPNITEYMFCAGYSDGSKDSCKGDSGGPHATHYRGTWYL
TGIVSWGQGCATVGHFGVYTRVSQYIEWLQKLMRSEPRPGVLLRAPFP

>1EAW|Chains A,C|SUPPRESSOR OF TUMORIGENICITY 14|HOMO SAPIENS(9606)
VVGGTDADEGEWPWQVSLHALGQGHICGASLISPNWLVSAAHCYIDDRGFR

YSDPTQWTAFLGLHDQSQRSAPGVQERRLKRIISHPFFNDFTFDYDIALLELE
KPAEYSSMVRPICLPDASHVFPAGKAIWVTGWGHTQYGGTGALILQKGEIRV
INQTTCENLLPQQITPRMMCVGFLSGGVDSCQGDSGGPLSSVEADGRIFQAG
VVSWGDGCAQRNKPGVYTRLPLFRDWIKENTGV

>1C08|Chain B|ANTI-HEN EGG WHITE LYSOZYME ANTIBODY (HYHEL-10)|Mus musculus(10090)
DVQLQESGPSLVKPSQTLSLTCSVTGDSITSDYWSWIRKFPGNRLEYMGYVS
YSGSTYYNPSLKSRISITRDTSKNQYYLDLNSVTTEDTATYYCANWDGDYWG
QGTLVTVSAA

>1C08|Chain A|ANTI-HEN EGG WHITE LYSOZYME ANTIBODY (HYHEL-10)|Mus musculus(10090)
DIVLTQSPATLSVTPGNSVSLSCRASQSIGNNLHWYQQKSHESPRLLIKYASQSI
SGIPSRFSGSGSGTDFTLSINSVETEDFGMYFCQQSNSWPYTFGGGTKLEIK

>1DAN|Chain T|SOLUBLE TISSUE FACTOR|Homo sapiens (9606)
NTVAAYNLTWKSTNFKTILEWEPKPVNQVYTVQISTKSGDWKSKCFYTTDT
ECDLTDEIVKDVKQTYLARVFSYPAGNVE

>1A22|Chain B|GROWTH HORMONE RECEPTOR|Homo sapiens (9606)
FSGSEATAAILSRAPWSLQSVNPGLKTNSSKEPKFTKCRSPERETFSCHWTDE
VHHGTKNLGPIQLFYTRRNTQEWTQEWKECPDYVSAGENSCYFNSSFTSIWI
PYCIKLTSNGGTVDEKCFSVDEIVQPDPPIALNWTLLNVSLTGIHADIQVRWE
APRNADIQKGWMVLEYELQYKEVNETKWKMMDPILTTSVPVYSLKVDKEY
EVRVRSKQRNSGNYGEFSEVLYVTLPQMSQ

>1IAR|Chain B|PROTEIN (INTERLEUKIN-4 RECEPTOR ALPHA CHAIN)|Homo sapiens(9606)
FKVLQEPTCVSDYMSISTCEWKMNGPTNCSTELRLLYQLVFLLSEAHTCIPEN
NGGAGCVCHLLMDDVVSADNYTLDLWAGQQLLWKGSFKPSEHVKPRAPGN
LTVHTNVSDTLLLTWSNPYPPDNYLYNHLTYAVNIWSENDPADFRIYNVTYL
EPSLRIAASTLKSGISYRARVRAWAQAYNTTWSEWSPSTKWHNSYREPFEQH

>1GC1|Chain C|CD4|Homo sapiens(9606)
KKVVLGKKGDTVELTCTASQKKSIQFHWKNSNQIKILGNQGSFLTKGPSKLN
DRADSRRSLWDQGNFPLIIKNLKIEDSDTYICEVEDQKEEVQLLVFGLTANSD
THLLQGQSLTLTLESPPGSSPSVQCRSPRGKNIQGGKTLSVSQLELQDSGTWT
CTVLQNQKKVEFKIDIVVLAFQKASNT

>1A22|Chain A|GROWTH HORMONE|Homo sapiens(9606)
FPTIPLSRLFDNAMLRAHRLHQLAFDTYQEFEEAYIPKEQKYSFLQNPQTSLC
FSESIPTPSNREETQQKSNLELLRISLLLIQSWLEPVQFLRSVFANSLVYGASDS
NVYDLLKDLEERIQTLMGRLEDGSPRTGQIFKQTYSKFDTNSHNDDALLKNY

GLLYCFRKDMDKVETFLRIVQCRSVEGSCGF

>1JRH|Chain H|ANTIBODY A6|Mus musculus(10090)
AVKLQESGPGILKPSQTLSLTCSFSGFSLTTYGMGVGWIRQSSGKGLEWLAH
IWWDDDKYYNPSLKSRLTISKDTSRNQVFLKITSVATADTATYYCARRAPFY
GNHAMDYWGQGTTVTVSSAKTTPPSVYPLAPGSAAQTNSMVTLGCLVKGY
FPEPVTVTWNSGSLSSGVHTFPAVLQSDLYTLSSSVTVPSSPRPSETVTCNVA
HPASSTKVDKKI

>1JRH|Chain L|ANTIBODY A6|Mus musculus(10090)
SVEMTQSPSSFSVSLGDRVTITCKASEDIYNRLAWYQQKPGNAPRLLISGATS
LETEVPSRFSGSGSGKDYTLSITSLQTEDVATYYCQQYWSTWTFGGGTKLEI
KRADAAPTVSIFPPSSEQLTSGGASVVCFLNNFYPKDINVKWKIDGSERQNG
VLNSWTDQDSKDSTYSMSSTLTLTKDEYERHNSYTCEATHKTSTSPIVKSFN
RNEC

>1JTG|Chains B,D|BETA-LACTAMASE INHIBITORY PROTEIN|Streptomyces clavuligerus(1901)
AGVMTGAKFTQIQFGMTRQQVLDIAGAENCETGGSFGDSIHCRGHAAGDYY
AYATFGFTSAAADAKVDSKSQEKLLAPSAPTLTLAKFNQVTVGMTRAQVLA
TVGQGSCTTWSEYYPAYPSTAGVTLSLSCFDVDGYSSTGFYRGSAHLWFTD
GVLQGKRQWDLV

>1AK4|Chains C,D|HIV-1 CAPSID|Human immunodeficiency virus 1(11676)
PIVQNLQGQMVHQAISPRTLNAWVKVVEEKAFSPEVIPMFSALSEGATPQDL
NTMLNTVGGHQAAMQMLKETINEEAAEWDRLHPVHAGPIAPGQMREPRGS
DIAGTTSTLQEQIGWMTHNPPIPVGEIYKRWIILGLNKIVRMY

>2O3B|Chain B|Sugar-non-specific nuclease inhibitor|Nostoc sp.(103690)
GSTKTNSEILEQLKQASDGLLFMSESEYPFEVFLWEGSAPPVTHEIVLQQTGH
GQDAPFKVVDIDSFFSRATTPQDWYEDEENAVVAKFQKLLEVIKSNLKNPQV
YRLGEVELDVYVIGETPAGNLAGISTKVVET

>1DAN|Chain L|BLOOD COAGULATION FACTOR VIIA light chain|Homo sapiens(9606)
ANAFLEELRPGSLERECKEEQCSFEEAREIFKDAERTKLFWISYSDGDQCASS
PCQNGGSCKDQLQSYICFCLPAFEGRNCETHKDDQLICVNENGGCEQYCSDH
TGTKRSCRCHEGYSLLADGVSCTPTVEYPCGKIPILEKRNASKPQGR

>1EMV|Chain B|COLICIN E9|Escherichia coli(562)
MESKRNKPGKATGKGKPVGDKWLDDAGKDSGAPIPDRIADKLRDKEFKSF
DDFRKAVWEEVSKDPELSKNLNPSNKSSVSKGYSPFTPKNQQVGGRKVYEL
HHDKPISQGGEVYDMDNIRVTTPKRHIDIHRGK

>3HFM|Chain Y|HEN EGG WHITE LYSOZYME|Gallus gallus (9031)
KVFGRCELAAAMKRHGLDNYRGYSLGNWVCAAKFESNFNTQATNRNTDGS
TDYGILQINSRWWCNDGRTPGSRNLCNIPCSALLSSDITASVNCAKKIVSDGN
GMNAWVAWRNRCKGTDVQAWIRGCRL

>1IAR1|Chain A|PROTEIN (INTERLEUKIN-4)|Homo sapiens (9606)
HKCDITLQEIIKTLNSLTEQKTLCTELTVTDIFAASKNTTEKETFCRAATVLR
QFYSHHEKDTRCLGATAQQFHRHKQLIRFLKRLDRNLWGLAGLNSCPVKEA
NQSTLENFLERLKTIMREKYSKCSS

>1H9D|Chains B,D|CORE-BINDING FACTOR CBF-BETA|HOMO SAPIENS(9606)
PRVVPDQRSKFENEEFFRKLSRECEIKYTGFRDRPHEERQARFQNACRDGRS
EIAFVATGTNLSLQFFPASWQGEQRQTPSREYVDLEREAGKVYLKAPMILNG
VCVIWKGWIDLQRLDGMGCLEFDEERAQQE

>2J0T|Chains D,E,F|METALLOPROTEINASE INHIBITOR 1|HOMO SAPIENS(9606)
CTCVPPHPQTAFCNSDLVIRAKFVGTPEVNQTTLYQRYEIKMTKMYKGFQA
LGDAADIRFVYTPAMESVCGYFHRSHNRSEEFLIAGKLQDGLLHITTCSFVAP
WNSLSLAQRRGFTKTYTVGCEE

>1A4Y|Chains B,E|ANGIOGENIN|Homo sapiens(9606)
QDNSRYTHFLTQHYDAKPQGRDDRYCESIMRRRGLTSPCKDINTFIHGNKRS
IKAICENKNGNPHRENLRISKSSFQVTTCKLHGGSPWPPCQYRATAGFRNVV
VACENGLPVHLDQSIFRRP

>1DVF|Chain B|FV D1.3|Mus musculus(10090)
QVQLQESGPGLVAPSQSLSITCTVSGFSLTGYGVNWVRQPPGKGLEWLGMI
WGDGNTDYNSALKSRLSISKDNSKSQVFLKMNSLHTDDTARYYCARERDYR
LDYWGQGTTLTVSS

>1NMB|Chain H|FAB NC10|Mus musculus(10090)
QVQLQQPGAELVKPGASVRMSCKASGYTFTNYNMYWVKQSPGQGLEWIGI
FYPGNGDTSYNQKFKDKATLTADKSSNTAYMQLSSLTSEDSAVYYCARSGGS
YRYDGGFDYWGQGTTLTVSS

>1DVF|Chain D|FV E5.2|Mus musculus(10090)
QVQLQQSGTELVKSGASVKLSCTASGFNIKDTHMNWVKQRPEQGLEWIGRI
DPANGNIQYDPKFRGKATITADTSSNTAYLQLSSLTSEDTAVYYCATKVIYY
QGRGAMDYWGQGTTLTVS

>1BRS|Chains A,B,C|BARNASE|Bacillus amyloliquefaciens(1390)
AQVINTFDGVADYLQTYHKLPDNYITKSEAQALGWVASKGNLADVAPGKSI

GGDIFSNREGKLPGKSGRTWREADINYTSGFRNSDRILYSSDWLIYKTTDHY
QTFTKIR

>1DVF|Chain A|FV D1.3|Mus musculus (10090)
DIVLTQSPASLSASVGETVTITCRASGNIHNYLAWYQQKQGKSPQLLVYYTTT
LADGVPSRFSGSGSGTQYSLKINSLQPEDFGSYYCQHFWSTPRTFGGGTKLEI
KR

>1KTZ|Chain B|TGF-beta Type II Receptor|Homo sapiens(9606)
VTDNNGAVKFPQLCKFCDVRFSTCDNQKSCMSNCSITSICEKPQEVCVAVWR
KNDENITLETVCHDPKLPYHDFILEDAASPKCIMKEKKKPGETFFMCSCSSDE
CNDNIIFSEEYNTSNPD

>1JRH|Chain I|INTERFERON-GAMMA RECEPTOR ALPHA CHAIN|Homo sapiens(9606)
EMGTADLGPSSVPTPTNVTIESYNMNPIVYWEYQIMPQVPVFTVEVKNYGV
KNSEWIDACINISHHYCNISDHVGDPSNSLWVRVKARVGQKESAYAKSEEFA
VSRDG

>1BRS|Chains D,E,F|BARSTAR|Bacillus amyloliquefaciens(1390)
KKAVINGEQIRSISDLHQTLKKELALPEYYGENLDALWDALTGWVEYPLVLE
WRQFEQSKQLTENGAESVLQVFREAKAEGADITIILS

>2JEL|Chain P|HISTIDINE-CONTAINING PROTEIN|Escherichia coli (562)
MFQQEVTITAPNGLHTRPAAQFVKEAKGFTSEITVTSNGKSASAKSLFKLQT
LGLTQGTVVTISAEGEDEQKAVEHLVKLMAELE

>1BXI|Chain A|PROTEIN (COLICIN E9 IMMUNITY PROTEIN)|Escherichia coli(562)
MELKASISDYTEAEFLQLVTTICNADTSSEEELVKLVTHFEEMTEHPSGSDLIY
YPKEGDDDSPSGIVNTVQQWRAANGKSGFKQG

>1KTZ|Chain A|TRANSFORMING GROWTH FACTOR BETA 3|Homo sapiens(9606)
ALDTNYCFRNLEENCCVRPLYIDFRQDLGWKWVHEPKGYYANFCSGPCPY
LRSADTTHSTVLGLYNTLNPEASASPCCVPQDLEPLTILYYVGRTPKVEQLSN
MVVKSCKCS

>2WPT|Chain A|COLICIN-E2 IMMUNITY PROTEIN|ESCHERICHIA COLI (562)
MELKHSISDYTEAEFLEFVKKIARAEGATECDDNKLVREFERLTEHPDGSDLI
YYPRDDREDSPEGIVKEIKEWRAANGKSGFKQG

>1XD3|Chains B,D|UBC protein|Homo sapiens(9606)
MQIFVKTLTGKTITLEVEPSDTIENVKAKIQDKEGIPPDQQRLIFAGKQLEDG
RTLSDYNIQKESTLHLVLRLRG

>1FFW|Chains B,D|CHEMOTAXIS PROTEIN CHEA|Escherichia coli(562)
RQLALEAKGETPSAVTRLSVVAKSEPQDEQSRSQSARRIILSRLKAGEVDLLE
EELGHLTTLTDVVKGADSLSAILPGDIAEDDITAVLCFVIEADQITFETVEVSP
KISTPPVLKLAAEQAPTGRVEREKTTR

>1TM1|Chain I|chymotrypsin inhibitor 2|Hordeum vulgare subsp. vulgare(112509)
MKTEWPELVGKSVEEAKKVILQDKPAAQIIVLPVGTIVTMEYRIDRVRLFVD
RLDNIAQVPRVG

>1CBW|Chains D,I|BPTI|Bos taurus (9913)
RPDFCLEPPYTGPCKARIIRYFYNAKAGLCQTFVYGGCRAKRNNFKSAEDC
MRTCGGA

>1FCC|Chains C,D|STREPTOCOCCAL PROTEIN G| >(C2 FRAGMENT) Streptococcus(1301)
TTYKLVINGKTLKGETTTEAVDAATAEKVFKQYANDNGVDGEWTYDDATK
TFTVTE

>1CHO|Chain I|TURKEY OVOMUCOID THIRD DOMAIN (OMTKY3)|Meleagris gallopavo(9103)
LAAVSVDCSEYPKPACTLEYRPLCGSDNKTYGNKCNFCNAVVESNGTLTLSH
FGKC

>1FC2|Chain C|FRAGMENT B OF PROTEIN A COMPLEX|Staphylococcus aureus subsp. aureus (93061)
ADNKFNKEQQNAFYEILHLPNLNEEQRNGFIQSLKDDPSQSANLLAEAKKLN
DAQXXK

>1F47|Chain A|CELL DIVISION PROTEIN FTSZ|Escherichia coli(562)
KEPDYLDIPAFLRKQAD

>1DN2|Chains E,F|ENGINEERED PEPTIDE|null
DCAWHLGELVWCTX

7.14 FASTA sequences for residues in BID-18

>1CDL|Chains A,B,C,D|CALMODULIN|Homo sapiens(9606)
ADQLTEEQIAEFKEAFSLFDKDGDGTITTKELGTVMRSLGQNPTEAELQDMI
NEVDADGNGTIDFPEFLTMMARKMKDTDSEEEIREAFRVFDKDGNGYISAA
ELRHVMTNLGEKLTDEEVDEMIREADIDGDGQVNYEEFVQMMTA

>1CDL|Chains E,F,G,H|CALCIUM/CALMODULIN-DEPENDENT PROTEIN KINASE TYPE II ALPHA CHAIN|
ARRKWQKTGHAVRAIGRLSS

>1DVA|Chains H,I|DES-GLA FACTOR VIIA (HEAVY CHAIN)|Homo sapiens(9606)
IVGGKVCPKGECPWQVLLLVNGAQLCGGTLINTIWVVSAAHCFDKIKNWRN
LIAVLGEHDLSEHDGDEQSRRVAQVIIPSTYVPGTTNHDIALLRLHQPVVLTD
HVVPLCLPERTFSERTLAFVRFSLVSGWGQLLDRGATALELMVLNVPRLMT
QDCLQQSRKVGDSPNITEYMFCAGYSDGSKDSCKGDSGGPHATHYRGTWYL
TGIVSWGQGCATVGHFGVYTRVSQYIEWLQKLMRSEPRPGVLLRAPFP

>1DVA|Chains X,Y|PEPTIDE E-76|null XALCDDPRVDRWYCQFVEGX

>1DX5|Chains M,N,O,P|Thrombin heavy chain|Homo sapiens(9606) IV
EGSDAEIGMSPWQVMLFRKSPQELLCGASLISDRWVLTAAHCLLYPPWDKN
FIENDLLVRIGKHSRTRYERNIEKISMLEKIYIHPRYNWRENLDRDIALMKLKK
PVAFSDYIHPVCLPDRETAASLLQAGYKGRVTGWGNLKETWTANVGKGQP
SVLQVVNLPIVERPVCKDSTRIRITDNMFCAGYKPDEGKRGDACEGDSGGPF
VMKSPFNNRWYQMGIVSWGEGCDRDGKYGFYTHVFRLKKWIQKVIDQFGE

>1EBP|Chains A,B|EPO RECEPTOR|Homo sapiens(9606) KFESKAAL
LAARGPEELLCFTERLEDLVCFWEEAASAGVGPGNYSFSYQLEDEPWKLCR
LHQAPTARGAVRFWCSLPTADTSSFVPLELRVTAASGAPRYHRVIHINEVVL
LDAPVGLVARLADESGHVVLRWLPPPETPMTSHIRYEVDVSAGNGAGSVQR
VEILEGRTECVLSNLRGRTRYTFAVRARMAEPSFGGFWSAWSEPVSLLT

>1EBP|Chains C,D|EPO MIMETICS PEPTIDE 1|
GGTYSCHFGPLTWVCKPQGG

>1ES7|Chains A,C|BONE MORPHOGENETIC PROTEIN-2|Homo sapiens(9606)
MAQAKHKQRKRLKSSCKRHPLYVDFSDVGWNDWIVAPPGYHAFYCHGECP
FPLADHLNSTNHAIVQTLVNSVNSKIPKACCVPTELSAISMLYLDENEKVVLK
NYQDMVVEGCGCR

>1FAK|Chain T|PROTEIN (SOLUBLE TISSUE FACTOR)|Homo sapiens(9606)
NTVAAYNLTWKSTNFKTILEWEPKPVNQVYTVQISTKSGDWKSKCFYTTDT
ECDLTDEIVKDVKQTYLARVFSYPAGNVESTGSAGEPLYENSPEFTPYLETN

LGQPTIQSFEQVGTKVNVTVEDERTLVRRNNTFLSLRDVFGKDLIYTLYYWK
SSSSGKKTAKTNTNEFLIDVDKGENYCFSVQAVIPSRTVNRKSTDSPVECM

>1FE8|Chains A,B,C|VON WILLEBRAND FACTOR|Homo sapiens(9606)
GSHMAPDCSQPLDVILLLDGSSSFPASYFDEMKSFAKAFISKANIGPRLTQVSV
LQYGSITTIDVPWNVVPEKAHLLSLVDVMQREGGPSQIGDALGFAVRYLTSE
MHGARPGASKAVVILVTDVSVDSVDAAADAARSNRVTVFPIGIGDRYDAAQL
RILAGPAGDSNVVKLQRIEDLPTMVTLGNSFLHKLCSG

>1FOE|Chains B,D,F,H|RAS-RELATED C3 BOTULINUM TOXIN SUBSTRATE|Homo sapiens(9606)
MQAIKCVVVGDGAVGKTCLLISYTTNAFPGEYIPTVFDNYSANVMVDGKPV
NLGLWDTAGQEDYDRLRPLSYPQTDVFLICFSLVSPASFENVRAKWYPEVRH
HCPNTPIILVGTKLDLRDDKDTIEKLKEKKLTPITYPQGLAMAKEIGAVKYLE
CSALTQRGLKTVFDEAIRAVL

>1G3I|Chains G,H,I,J,K,L,M,N,O,P,Q,R|ATP-DEPENDENT PROTEASE HSLV|Haemophilus influenzae(727)
TTIVSVRRNGQVVVGGDGQVSLGNTVMKGNARKVRRLYNGKVLAGFAGGT
ADAFTLFELFERKLEMHQGHLLKSAVELAKDWRTDRALRKLEAMLIVADEK
ESLIITGIGDVVQPEEDQILAIGSGGNYALSAARALVENTELSAHEIVEKSLRIA
GDICVFTNTNFTIEELPN

>1GL4|Chain A|NIDOGEN-1|MUS MUSCULUS(10090)
APLAQQTCANNRHQCSVHAECRDYATGFCCRCVANYTGNGRQCVAEGSPQ
RVNGKVKGRIFVGSSQVPVVFENTDLHSYVVMNHGRSYTAISTIPETVGYSLL
PLAPIGGIIGWMFAVEQDGFKNGFSITGGEFTRQAEVTFLGHPGKLVLKQQF
SGIDEHGHLTISTELEGRVPQIPYGASVHIEPYTELYHYSSSVITSSSTREYTVM
EPDQDGAAPSHTHIYQWRQTITFQECAHDDARPALPSTQQLSVDSVFVLYNK
EERILRYALSNSIGPVRDGSPDA

>1IHB|Chains A,B|CYCLIN-DEPENDENT KINASE 6 INHIBITOR|Homo sapiens(9606)
MAEPWGNELASAAARGDLEQLTSLLQNNVNVNAQNGFGRTALQVMKLGNP
EIARRLLLRGANPDLKDRTGFAVIHDAARAGFLDTLQTLLEFQADVNIEDNE
GNLPLHLAAKEGHLRVVEFLVKHTASNVGHRNHKGDTACDLARLYGRNEVV
SLMQANGAG

>1JAT|Chain A|Ubiquitin-Conjugating Enzyme E2-17.5 KDA|Saccharomyces cerevisiae(4932)
GSAASLPKRIIKETEKLVSDPVPGITAEPHDDNLRYFQVTIEGPEQSPYEDGIF
ELELYLPDDYPMEAPKVRFLTKIYHPNIDRLGRICLDVLKTNWSPALQIRTVL
LSIQALLASPNPNDPLANDVAEDWIKNEQGAKAKAREWTKLYAKKKPE

>1JAT|Chain B|Ubiquitin-Conjugating Enzyme Variant Mms2|Saccharomyces cerevisiae(4932)
HMSKVPRNFRLLEELEKGEKGFGPESCSYGLADSDDITMTKWNGTILGPPHS
NHENRIYSLSIDCGPNYPDSPPKVTFISKINLPCVNPTTGEVQTDFHTLRDWK
RAYTMETLLLDLRKEMATPANKKLRQPKEGETF

>1JPP|Chains A,B|BETA-CATENIN|Mus musculus(10090)
HAVVNLINYQDDAELATRAIPELTKLLNDEDQVVVNKAAVMVHQLSKKEAS
RHAIMRSPQMVSAIVRTMQNTNDVETARCTAGTLHNLSHHREGLLAIFKSGG
IPALVKMLGSPVDSVLFYAITTLHNLLLHQEGAKMAVRLAGGLQKMVALLNK
TNVKFLAITTDCLQILAYGNQESKLIILASGGPQALVNIMRTYTYEKLLWTTS
RVLKVLSVCSSNKPAIVEAGGMQALGLHLTDPSQRLVQNCLWTLRNLSDAAT
KQEGMEGLLGTLVQLLGSDDINVVTCAAGILSNLTCNNYKNKMMVCQVGGI
EALVRTVLRAGDREDITEPAICALRHLTSRHQEAEMAQNAVRLHYGLPVVVK
LLHPPSHWPLIKATVGLIRNLALCPANHAPLREQGAIPRLVQLLVRAHQDTQ
RRTSMGGTQQQFVEGVRMEEIVEGCTGALHILARDVHNRIVIRGLNTIPLFV
QLLYSPIENIQRVAAGVLCELAQDKEAAEAIEAEGATAPLTELLHSRNEGVAT
YAAAVLFRMSEDKPQDYK

>1MQ8|Chains B,D|Integrin alpha-L|Homo sapiens (9606)
VDLVFLFDGSMSLQPDEFQKILDFMKDVMKKCSNTSYQFAAVQFSTSYKTEF
DFSDYVKRKDPDALLKHVKHMLLLTNTFGAINYVATEVFREELGARPDATK
VLIIITDGEATDSGNIDAAKDIIRYIIGIGKHFQTKESQETLHKFASKPASEFVKI
LDTFEKLKDLCTELQKKI

>1NFI|Chains E,F|I-KAPPA-B-ALPHA|Homo sapiens (9606)
LTEDGDSFLHLAIIHEEKALTMEVIRQVKGDLAFLNFQNNLQQTPLHLAVITN
QPEIAEALLGAGCDPELRDFRGNTPLHLACEQGCLASVGVLTQSCTTPHLHSI
LKATNYNGHTCLHLASIHGYLGIVELLVSLGADVNAQEPCNGRTALHLAVDL
QNPDLVSLLLKCGADVNRVTYQGYSPYQLTWGRPSTRIQQQLGQLTLENLQ
MLPE

>1NUN|Chain A|Fibroblast growth factor-10|Homo sapiens(9606) GRH
VRSYNHLQGDVRWRKLFSFTKYFLKIEKNGKVSGTKKENCPYSILEITSVEIG
VVAVKAINSNYYLAMNKKGKLYGSKEFNNDCKLKERIEENGYNTYASFNWQ
HNGRQMYVALNGKGAPRRGQKTRRKNTSAHFLPMVVHS

>1UB4|Chain C|MazE protein|Escherichia coli(562)
GPHMIHSSVKRWGNSPAVRIPATLMQALNLNIDDEVKIDLVDGKLIIEPVRKE
PVFTLAELVNDITPENLHENIDWGEPKDKEVW

>2HHB|Chains B,D|HEMOGLOBIN (DEOXY) (BETA CHAIN)|Homo sapiens(9606)
VHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTP

DAVMGNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPE
NFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH

Bibliography

[Alt+97] Stephen F Altschul et al. "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs". In: *Nucleic acids research* 25.17 (1997), pp. 3389–3402.

[Ara+14] Aleksandr Aravkin et al. "A variational approach to stable principal component pursuit". In: *arXiv preprint arXiv:1406.1089* (2014).

[Ash+10] Haim Ashkenazy et al. "ConSurf 2010: calculating evolutionary conservation in sequence and structure of proteins and nucleic acids". In: *Nucleic acids research* 38.suppl_2 (2010), W529–W533.

[Ass+10] Salam A Assi et al. "PCRPi: Presaging Critical Residues in Protein interfaces, a new computational tool to chart hot spots in protein interfaces". In: *Nucleic acids research* 38.6 (2010), e86–e86.

[BD98] Kristin Bennett and Ayhan Demiriz. "Semi-supervised support vector machines". In: *Advances in Neural Information processing systems* 11 (1998).

[Ben+12] Dennis A Benson et al. "GenBank". In: *Nucleic acids research* 41.D1 (2012), pp. D36–D42.

[Ber+00a] Helen M Berman et al. "The protein data bank". In: *Nucleic acids research* 28.1 (2000), pp. 235–242.

[Ber+00b] Helen M Berman et al. "The protein data bank". In: *Nucleic acids research* 28.1 (2000), pp. 235–242.

[BGV92] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. "A training algorithm for optimal margin classifiers". In: *Proceedings of the fifth annual workshop on Computational learning theory*. 1992, pp. 144–152.

[Bis06] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[Boe+03] Brigitte Boeckmann et al. "The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003". In: *Nucleic acids research* 31.1 (2003), pp. 365–370.

[Bot+23] Patricia Mirela Bota et al. "CM2D3: Furnishing the human interactome with structural models of protein complexes derived by comparative modeling and docking". In: *Journal of Molecular Biology* (2023), p. 168055.

[Bre+09] Ryan Brenke et al. “Fragment-based identification of druggable ‘hot spots’ of proteins using Fourier domain correlation techniques”. In: *Bioinformatics* 25.5 (2009), pp. 621–627.

[Bre01] Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32.

[Bro+09] Christopher J Brown et al. “Awakening guardian angels: drugging the p53 pathway”. In: *Nature Reviews Cancer* 9.12 (2009), pp. 862–873.

[BT98] Andrew A Bogan and Kurt S Thorn. “Anatomy of hot spots in protein interfaces”. In: *Journal of molecular biology* 280.1 (1998), pp. 1–9.

[BTS02] J. M. Berg, J. L. Tymoczko, and L. Stryer. *Biochemistry 5th Edition*. W. H. Freeman, 2002.

[Bur19] Andriy Burkov. *The hundred-page machine learning book*. Vol. 1. Andriy Burkov Quebec City, QC, Canada, 2019.

[BZ14] Thierry Bouwmans and El Hadi Zahzah. “Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance”. In: *Computer Vision and Image Understanding* 122 (2014), pp. 22–34.

[Can+11] Emmanuel J Candès et al. “Robust principal component analysis?” In: *Journal of the ACM (JACM)* 58.3 (2011), pp. 1–37.

[Cap+98] Corinne Capoulade et al. “Overexpression of MDM2, due to enhanced translation, results in inactivation of wild-type p53 in Burkitt’s lymphoma cells”. In: *Oncogene* 16.12 (1998), pp. 1603–1610.

[CC82] Marvin Charton and Barbara I Charton. “The structural dependence of amino acid hydrophobicity parameters”. In: *Journal of theoretical biology* 99.4 (1982), pp. 629–644.

[CCS10] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. “A Singular Value Thresholding Algorithm for Matrix Completion”. In: *SIAM Journal on Optimization* 20.4 (2010), pp. 1956–1982.

[CG16] Tianqi Chen and Carlos Guestrin. “Xgboost: A scalable tree boosting system”. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[Cha] Nagesh Singh Chauhan. URL: `https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html`.

[CHW17] Chuming Chen, Hongzhan Huang, and Cathy H Wu. “Protein bioinformatics databases and resources”. In: *Protein bioinformatics: from protein modifications and networks to proteomics* (2017), pp. 3–39.

[CJ20] Davide Chicco and Giuseppe Jurman. “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation”. In: *BMC genomics* 21.1 (2020), pp. 1–13.

[CKL09a] Kyu-il Cho, Dongsup Kim, and Doheon Lee. "A feature-based approach to modeling protein–protein interaction hot spots". In: *Nucleic acids research* 37.8 (2009), pp. 2672–2687.

[CKL09b] Kyu-il Cho, Dongsup Kim, and Doheon Lee. "A feature-based approach to modeling protein–protein interaction hot spots". In: *Nucleic acids research* 37.8 (2009), pp. 2672–2687.

[CSZ09] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]". In: *IEEE Transactions on Neural Networks* 20.3 (2009), pp. 542–542.

[CSZ10] O. Chapelle, B. Scholkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 2010.

[CW89] Brian C Cunningham and James A Wells. "High-resolution epitope mapping of hGH-receptor interactions by alanine-scanning mutagenesis". In: *Science* 244.4908 (1989), pp. 1081–1085.

[Den+13] Lei Deng et al. "Boosting prediction performance of protein–protein interaction hot spots by using structural neighborhood properties". In: *Journal of Computational Biology* 20.11 (2013), pp. 878–891.

[DG17] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: `http://archive.ics.uci.edu/ml`.

[Din+13] Qingjie Ding et al. "Discovery of RG7388, a potent and selective p53–MDM2 inhibitor in clinical development". In: *Journal of medicinal chemistry* 56.14 (2013), pp. 5979–5983.

[DPM07] Steven J Darnell, David Page, and Julie C Mitchell. "An automated decision-tree approach to predicting protein interaction hot spots". In: *Proteins: Structure, Function, and Bioinformatics* 68.4 (2007), pp. 813–823.

[DS15] Alessia David and Michael JE Sternberg. "The contribution of missense mutations in core and rim residues of protein–protein interfaces to human disease". In: *Journal of molecular biology* 427.17 (2015), pp. 2886–2898.

[Eis+84] D Eisenberg et al. "Analysis of membrane and surface protein sequences with the hydrophobic moment plot". In: *Journal of molecular biology* 179.1 (1984), pp. 125–142.

[Eva+21] Richard Evans et al. "Protein complex prediction with AlphaFold-Multimer". In: *BioRxiv* (2021), pp. 2021–10.

[FGK13] James S Fraser, John D Gross, and Nevan J Krogan. "From systems to structure: bridging networks and mechanism". In: *Molecular cell* 49.2 (2013), pp. 222–231.

[FI04] Anis Feki and Irmgard Irminger-Finger. "Mutational spectrum of p53 mutations in primary breast and ovarian tumors". In: *Critical reviews in oncology/hematology* 52.2 (2004), pp. 103–116.

[Fis+03] TB Fischer et al. “The binding interface database (BID): a compilation of amino acid hot spots in protein interfaces”. In: *Bioinformatics* 19.11 (2003), pp. 1453–1454.

[Fis+05] Tiffany B Fischer et al. “A guide to protein interaction databases”. In: *The proteomics protocols handbook*. Springer, 2005, pp. 753–799.

[Fri01] Jerome H Friedman. “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics* (2001), pp. 1189–1232.

[FWL99] DA Freedman, L Wu, and AJ Levine. “Functions of the MDM2 oncoprotein”. In: *Cellular and Molecular Life Sciences CMLS* 55.1 (1999), pp. 96–107.

[GBC17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. “Deep learning (adaptive computation and machine learning series)”. In: *Cambridge Massachusetts* (2017), pp. 321–359.

[GE03] Isabelle Guyon and André Elisseeff. “An introduction to variable and feature selection”. In: *Journal of machine learning research* 3.Mar (2003), pp. 1157–1182.

[GF08] Solène Grosdidier and Juan Fernández-Recio. “Identification of hot-spot residues in protein-protein interactions by computational docking”. In: *BMC bioinformatics* 9.1 (2008), p. 447.

[GG19] Shivani Gupta and Atul Gupta. “Dealing with noise problem in machine learning data-sets: A systematic review”. In: *Procedia Computer Science* 161 (2019), pp. 466–474.

[GNS02] Raphael Guerois, Jens Erik Nielsen, and Luis Serrano. “Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations”. In: *Journal of molecular biology* 320.2 (2002), pp. 369–387.

[Gra74] Richard Grantham. “Amino acid difference formula to help explain protein evolution”. In: *science* 185.4154 (1974), pp. 862–864.

[Gre+22] Joe G Greener et al. “A guide to machine learning for biologists”. In: *Nature Reviews Molecular Cell Biology* 23.1 (2022), pp. 40–55.

[GS10] Mu Gao and Jeffrey Skolnick. “Structural space of protein–protein interfaces is degenerate, close to complete, and highly connected”. In: *Proceedings of the National Academy of Sciences* 107.52 (2010), pp. 22517–22522.

[Guy+02] Isabelle Guyon et al. “Gene selection for cancer classification using support vector machines”. In: *Machine learning* 46.1 (2002), pp. 389–422.

[Ham05] Thomas Hamelryck. “An amino acid has two sides: a new 2D measure provides a different view of solvent exposure”. In: *Proteins: Structure, Function, and Bioinformatics* 59.1 (2005), pp. 38–48.

[Har+20] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[Has+09] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[HG15] Zena M Hira and Duncan F Gillies. "A review of feature selection and feature extraction methods applied on microarray data". In: *Advances in bioinformatics* 2015 (2015).

[HH92] Steven Henikoff and Jorja G Henikoff. "Amino acid substitution matrices from protein blocks." In: *Proceedings of the National Academy of Sciences* 89.22 (1992), pp. 10915–10919.

[HMK02] Shuanghong Huo, Irina Massova, and Peter A Kollman. "Computational alanine scanning of the 1: 1 human growth hormone–receptor complex". In: *Journal of computational chemistry* 23.1 (2002), pp. 15–27.

[HT93] SJ Hubbard and JM Thornton. "Naccess: Department of biochemistry and molecular biology, university college london". In: *Software available at http://www. bioinf. manchester. ac. uk/naccess/nacdownload. html* (1993).

[Hua+10] Ying Huang et al. "CD-HIT Suite: a web server for clustering and comparing biological sequences". In: *Bioinformatics* 26.5 (2010), pp. 680–682.

[Hun07] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[HW81] Thomas P Hopp and Kenneth R Woods. "Prediction of protein antigenic determinants from amino acid sequences." In: *Proceedings of the National Academy of Sciences* 78.6 (1981), pp. 3824–3828.

[Jan+19] Justina Jankauskaitė et al. "SKEMPI 2.0: an updated benchmark of changes in protein–protein binding energy, kinetics and thermodynamics upon mutation". In: *Bioinformatics* 35.3 (2019), pp. 462–469.

[Jan+78] Joel Janin et al. "Conformation of amino acid side-chains in proteins". In: *Journal of molecular biology* 125.3 (1978), pp. 357–386.

[Jan95] Joel Janin. "Elusive affinities". In: *Proteins: Structure, Function, and Bioinformatics* 21.1 (1995), pp. 30–39.

[JO99] Tamar Juven-Gershon and Moshe Oren. "Mdm2: the ups and downs". In: *Molecular medicine* 5.2 (1999), pp. 71–83.

[Jol02] Ian T Jolliffe. *Principal component analysis for special types of data*. Springer, 2002.

[Joo+10] Robbie P Joosten et al. “A series of PDB related databases for everyday needs”. In: *Nucleic acids research* 39.suppl_1 (2010), pp. D411–D419.

[JT96] Susan Jones and Janet M Thornton. “Principles of protein-protein interactions.” In: *Proceedings of the National Academy of Sciences* 93.1 (1996), pp. 13–20.

[JT97] Susan Jones and Janet M Thornton. “Analysis of protein-protein interaction sites using surface patches”. In: *Journal of molecular biology* 272.1 (1997), pp. 121–132.

[Kaw+07] Shuichi Kawashima et al. “AAindex: amino acid index database, progress report 2008”. In: *Nucleic acids research* 36.suppl_1 (2007), pp. D202–D205.

[KB02] Tanja Kortemme and David Baker. “A simple physical model for binding energy hot spots in protein–protein complexes”. In: *Proceedings of the National Academy of Sciences* 99.22 (2002), pp. 14116–14121.

[KB19] Dmitry Kobak and Philipp Berens. “The art of using t-SNE for single-cell transcriptomics”. In: *Nature communications* 10.1 (2019), pp. 1–14.

[KC21] Andrew J Kavran and Aaron Clauset. “Denoising large-scale biological data using network filters”. In: *BMC bioinformatics* 22 (2021), pp. 1–21.

[Kes+08] Ozlem Keskin et al. “Principles of protein- protein interactions: what are the preferred ways for proteins to interact?” In: *Chemical reviews* 108.4 (2008), pp. 1225–1244.

[KG06] MD Shaji Kumar and M Michael Gromiha. “PINT: protein–protein interactions thermodynamic database”. In: *Nucleic acids research* 34.suppl_1 (2006), pp. D195–D198.

[KJ97] Ron Kohavi and George H John. “Wrappers for feature subset selection”. In: *Artificial intelligence* 97.1-2 (1997), pp. 273–324.

[Kus+96] Paul H Kussie et al. “Structure of the MDM2 oncoprotein bound to the p53 tumor suppressor transactivation domain”. In: *Science* 274.5289 (1996), pp. 948–953.

[LEA15] P. Li, E. A. Stuart EA, and D. B. Allison. “Multiple Imputation: A Flexible Tool for Handling Missing Data”. In: *Journal of American Medical Association (JAMA)* 314.18 (2015), pp. 1966–1967.

[Lei+09] Christian Leistner et al. “Semi-supervised random forests”. In: *2009 IEEE 12th international conference on computer vision*. IEEE. 2009, pp. 506–513.

[Les10] Arthur Lesk. *Introduction to protein science: architecture, function, and genomics*. Oxford university press, 2010.

[Lev10] Emmanuel D Levy. “A simple definition of structural regions in proteins and its use in analyzing interface evolution”. In: *Journal of molecular biology* 403.4 (2010), pp. 660–670.

[Li+22] Shiwei Li et al. “Recent advances in predicting protein–protein interactions with the aid of artificial intelligence algorithms”. In: *Current Opinion in Structural Biology* 73 (2022), p. 102344.

[LLD18] Siyu Liu, Chuyao Liu, and Lei Deng. “Machine learning approaches for protein–protein interaction hot spot prediction: Progress and comparative assessment”. In: *Molecules* 23.10 (2018), p. 2535.

[LP85] David J Lipman and William R Pearson. “Rapid and sensitive protein similarity searches”. In: *Science* 227.4693 (1985), pp. 1435–1441.

[Mal+08] Pavan Kumar Mallapragada et al. “Semiboost: Boosting for semi-supervised learning”. In: *IEEE transactions on pattern analysis and machine intelligence* 31.11 (2008), pp. 2000–2014.

[McD09] Gary C McDonald. “Ridge regression”. In: *Wiley Interdisciplinary Reviews: Computational Statistics* 1.1 (2009), pp. 93–100.

[McK+10] Wes McKinney et al. “Data structures for statistical computing in python”. In: *Proceedings of the 9th Python in Science Conference*. Vol. 445. Austin, TX. 2010, pp. 51–56.

[MF12] Iain H Moal and Juan Fernández-Recio. “SKEMPI: a Structural Kinetic and Energetic database of Mutant Protein Interactions and its use in empirical models”. In: *Bioinformatics* 28.20 (2012), pp. 2600–2607.

[MFR07] Irina S Moreira, Pedro A Fernandes, and Maria J Ramos. “Hot spots—A review of the protein–protein interface determinant amino-acid residues”. In: *Proteins: Structure, Function, and Bioinformatics* 68.4 (2007), pp. 803–812.

[MG15] C. MOLNAR and J. GAIR. *Concepts of Biology-1st Canadian Edition*. OpenStax College, 2015.

[MHM20] Leland McInnes, John Healy, and James Melville. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2020. arXiv: `1802.03426 [stat.ML]`.

[Mik+98] Sebastian Mika et al. “Kernel PCA and de-noising in feature spaces”. In: *Advances in neural information processing systems* 11 (1998).

[Mit17] T.M. Mitchell. *Machine Learning*. McGraw Hill series in computer science. McGraw Hill, 2017. ISBN: 9781259096952. URL: `https://books.google.de/books?id=ifdcswEACAAJ`.

[MK99] Irina Massova and Peter A Kollman. “Computational alanine scanning to probe protein- protein interactions: a novel approach to evaluate binding free energies”. In: *Journal of the American Chemical Society* 121.36 (1999), pp. 8133–8143.

[MKT01] Julie C Mitchell, Rex Kerr, and Lynn F Ten Eyck. “Rapid atomic density methods for molecular shape characterization”. In: *Journal of Molecular Graphics and Modelling* 19.3-4 (2001), pp. 325–330.

[Mom+98] Jamil Momand et al. "The MDM2 gene amplification database". In: *Nucleic acids research* 26.15 (1998), pp. 3453–3459.

[Moo+19] Kevin R Moon et al. "Visualizing structure and transitions in high-dimensional biological data". In: *Nature biotechnology* 37.12 (2019), pp. 1482–1492.

[Mor+17] Irina S Moreira et al. "SpotOn: high accuracy identification of protein-protein interface hot-spots". In: *Scientific reports* 7.1 (2017), p. 8007.

[MS] Vishal Morde and Venkat Anurag Setty. URL: `https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d`.

[Mur+17] Yoichi Murakami et al. "Network analysis and in silico prediction of protein–protein interactions with applications in drug discovery". In: *Current Opinion in Structural Biology* 44 (2017). Carbohydrates: A feast of structural glycobiology • Sequences and topology: Computational studies of protein-protein interactions, pp. 134–142.

[MWD00] Jamil Momand, Hsiao-Huei Wu, and Gargi Dasgupta. "MDM2—master regulator of the p53 tumor suppressor protein". In: *Gene* 242.1-2 (2000), pp. 15–29.

[MZ12] John Kenneth Morrow and Shuxing Zhang. "Computational prediction of hot spot residues". In: *Current pharmaceutical design* 18.9 (2012), p. 1255.

[NH19] Lan Huong Nguyen and Susan Holmes. "Ten quick tips for effective dimensionality reduction". In: *PLoS computational biology* 15.6 (2019), e1006907.

[NT03] Irene MA Nooren and Janet M Thornton. "Diversity of protein–protein interactions". In: *The EMBO journal* 22.14 (2003), pp. 3486–3492.

[OR03] Yanay Ofran and Burkhard Rost. "Analysing six types of protein–protein interfaces". In: *Journal of molecular biology* 325.2 (2003), pp. 377–387.

[PDB] PDB-101. URL: `https://cdn.rcsb.org/pdb101/learn/resources/what-is-a-protein/what-is-a-protein-pres.pdf`.

[Pea88] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

[Ped+a] F. Pedregosa et al. *Cross-validation: evaluating estimator performance*. URL: `https://scikit-learn.org/stable/modules/cross_validation.html`.

[Ped+b] F. Pedregosa et al. *Feature selection*. URL: `https://scikit-learn.org/stable/modules/feature_selection.html`.

[Ped+c] F. Pedregosa et al. *Preprocessing data*. URL: `https://scikit-learn.org/stable/modules/preprocessing.html`.

[Ped+d] F. Pedregosa et al. *sklearn.decomposition.PCA*. URL: `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html`.

[Ped+11a] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[Ped+11b] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[Pet+16a] Ioanna Petta et al. "Modulation of protein–protein interactions for the development of novel therapeutics". In: *Molecular Therapy* 24.4 (2016), pp. 707–718.

[Pet+16b] Ioanna Petta et al. "Modulation of Protein–Protein Interactions for the Development of Novel Therapeutics". In: *Molecular Therapy* 24.4 (2016), pp. 707–718. ISSN: 1525-0016. DOI: `https://doi.org/10.1038/mt.2015.214`.

[PLA15] Marharyta Petukh, Minghui Li, and Emil Alexov. "Predicting binding free energy change caused by point mutations with knowledge-modified MM/PBSA method". In: *PLoS computational biology* 11.7 (2015), e1004276.

[Rao+14] V Srinivasa Rao et al. "Protein-protein interaction detection: methods and analysis". In: *International journal of proteomics* 2014 (2014).

[RBM22] Nícia Rosário-Ferreira, Alexandre MJJ Bonvin, and Irina S Moreira. "Using machine-learning-driven approaches to boost hot-spot's knowledge". In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 12.5 (2022), e1602.

[RDG] RDGerroranal. URL: `https://Analysis%20of%20Errors:http://faculty.sites.uci.edu/chem11/files/2013/11/RDGerroranal.pdf`.

[RP15] Stuart J Russell and Norvig Peter. *Artificial Intelligence: A Modern Approach*. Pearson, 2015.

[RS94a] Burkhard Rost and Chris Sander. "Conservation and prediction of solvent accessibility in protein families". In: *Proteins: Structure, Function, and Bioinformatics* 20.3 (1994), pp. 216–226.

[RS94b] Burkhard Rost and Chris Sander. "Conservation and prediction of solvent accessibility in protein families". In: *Proteins: Structure, Function, and Bioinformatics* 20.3 (1994), pp. 216–226.

[Rud16] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747* (2016).

[Rye+16] Connie Rye et al. *Biology*. OpenStax, 2016.

[SAF10] Joan Segura Mora, Salam A Assi, and Narcis Fernandez-Fuentes. "Presaging critical residues in protein interfaces-web server (PCRPi-W): a web server to chart hot spots in protein interfaces". In: *PLoS One* 5.8 (2010), e12352.

[Sam59] Arthur L Samuel. "Some Studies in Machine Learning Using the Game of Checkers". In: *IBM Journal of research and development* 3.3 (1959), pp. 210–229.

[Sco+16] Duncan E Scott et al. "Small molecules, big targets: drug discovery faces the protein–protein interaction challenge". In: *Nature Reviews Drug Discovery* 15.8 (2016), p. 533.

[Sel] Ivan Selesnick. *A Derivation of the Soft-Thresholding Function*. URL: `https://eeweb.engineering.nyu.edu/iselesni/lecture_notes/SoftThresholding.pdf`.

[SIL07] Yvan Saeys, Inaki Inza, and Pedro Larranaga. "A review of feature selection techniques in bioinformatics". In: *bioinformatics* 23.19 (2007), pp. 2507–2517.

[Sit+21] Divya Sitani et al. "Robust principal component analysis-based prediction of protein-protein interaction hot spots". In: *Proteins: Structure, Function, and Bioinformatics* 89.6 (2021), pp. 639–647.

[Sit23] Divya Sitani. *RBHS_Sitani*. Version 1.0.0. Aug. 2023. URL: `https://github.com/Divya1205/RBHS_Sitani`.

[Son+08] Jiangning Song et al. "HSEpred: predict half-sphere exposure from protein sequences". In: *Bioinformatics* 24.13 (2008), pp. 1489–1497.

[SR15] Takaya Saito and Marc Rehmsmeier. "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets". In: *PloS one* 10.3 (2015).

[Ste+06] Sarah A Mueller Stein et al. "Principal components analysis: a review of its application on molecular dynamics data". In: *Annual Reports in Computational Chemistry* 2 (2006), pp. 233–261.

[Sti97] Wesley E Stites. "Protein- protein interactions: interface structure, binding thermodynamics, and mutational analysis". In: *Chemical reviews* 97.5 (1997), pp. 1233–1250.

[Str06] Gilbert Strang. *Linear algebra and its applications.* Belmont, CA: Thomson, Brooks/Cole, 2006.

[Str19] G. Strang. *Linear Algebra and Learning from Data*. Wellesley-Cambridge Press, 2019.

[Sun+14] Daqing Sun et al. "Discovery of AMG 232, a potent, selective, and orally bioavailable MDM2–p53 inhibitor in clinical development". In: *Journal of medicinal chemistry* 57.4 (2014), pp. 1454–1472.

[TB01] Kurt S Thorn and Andrew A Bogan. "ASEdb: a database of alanine mutations and their effects on the free energy of binding in protein interactions". In: *Bioinformatics* 17.3 (2001), pp. 284–285.

[Ten+09] Shaolei Teng et al. “Modeling effects of human single nucleotide polymorphisms on protein-protein interactions”. In: *Biophysical journal* 96.6 (2009), pp. 2178–2188.

[Tib11] Robert Tibshirani. “Regression shrinkage and selection via the lasso: a retrospective”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73.3 (2011), pp. 273–282.

[Val13] Leslie Valiant. *Probably approximately correct: natureõs algorithms for learning and prospering in a complex world*. Basic Books (AZ), 2013.

[VH08] Laurens Van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE.” In: *Journal of machine learning research* 9.11 (2008).

[VL02] Karen H Vousden and Xin Lu. “Live or let die: the cell’s response to p53”. In: *Nature Reviews Cancer* 2.8 (2002), pp. 594–604.

[Vu+13] Binh Vu et al. “Discovery of RG7112: a small-molecule MDM2 inhibitor in clinical development”. In: *ACS medicinal chemistry letters* 4.5 (2013), pp. 466–469.

[Wan+14] Shaomeng Wang et al. “SAR405838: an optimized inhibitor of MDM2–p53 interaction that induces complete and durable tumor regression”. In: *Cancer research* 74.20 (2014), pp. 5855–5865.

[WEG87] Svante Wold, Kim Esbensen, and Paul Geladi. “Principal component analysis”. In: *Chemometrics and intelligent laboratory systems* 2.1-3 (1987), pp. 37–52.

[Wel91] James A Wells. “[18] Systematic mutational analyses of protein-protein interfaces”. In: *Methods in enzymology*. Vol. 202. Elsevier, 1991, pp. 390–411.

[WM07] James A Wells and Christopher L McClendon. “Reaching for high-hanging fruit in drug discovery at protein–protein interfaces”. In: *Nature* 450.7172 (2007), pp. 1001–1009.

[WS05] Gabriel Waksman and Clare Sansom. “Introduction: Proteomics and protein-protein interactions: Biology, chemistry, bioinformatics, and drug design”. In: *Proteomics and Protein-Protein Interactions*. Springer, 2005, pp. 1–18.

[Wu+93] Xiangwei Wu et al. “The p53-mdm-2 autoregulatory feedback loop.” In: *Genes & development* 7.7a (1993), pp. 1126–1132.

[Xia+16] Junfeng Xia et al. “Predicting hot spots in protein interfaces based on protrusion index, pseudo hydrophobicity and electron-ion interaction pseudopotential features”. In: *Oncotarget* 7.14 (2016), p. 18065.

[Yan+23] Yu Yan et al. “MIND-S is a deep-learning prediction model for elucidating protein post-translational modifications in human diseases”. In: *Cell Reports Methods* 3.3 (2023).

[ZG09] X. Zhu and A.B. Goldberg. *Introduction to Semi-supervised Learning*. Morgan & Claypool, 2009.

[ZH05] Hui Zou and Trevor Hastie. "Regularization and variable selection via the elastic net". In: *Journal of the royal statistical society: series B (statistical methodology)* 67.2 (2005), pp. 301–320.

[Zha+15] Yujun Zhao et al. "Small-molecule inhibitors of the MDM2–p53 protein–protein interaction (MDM2 Inhibitors) in clinical trials for cancer treatment: miniperspective". In: *Journal of medicinal chemistry* 58.3 (2015), pp. 1038–1052.

[ZM11] Xiaolei Zhu and Julie C Mitchell. "KFC2: a knowledge-based hot spot prediction method based on interface solvation, atomic density, and plasticity features". In: *Proteins: Structure, Function, and Bioinformatics* 79.9 (2011), pp. 2671–2683.